National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait pàrvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

University of Alberta

Flexible Access Control in Broadband Communication Networks

by

Shuang Deng

A thesis
submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree
of Doctor of Philosophy

Department of Computing Science

Edmonton, Alberta
Fall 1992

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-77423-1

Canada

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: Shuang Deng
TITLE OF THESIS: Flexible Access Control in Broadband Communication Networks

DEGREE: Doctor of Philosophy

YEAR THIS DEGREE GRANTED: 1992

Permission is hereby granted to UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis not extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(Signed) . . . . . .

Permanent Address:
#4 - 11517 - 76 Avenue
Edmonton, Alberta
Canada T6G 0K6

Date: October 5, 1992

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled **Flexible Access Control in Broadband Communication Networks** submitted by **Shuang Deng** in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

U. M. Maydell

A. E. Kamal

P. Gburzyński

B. Schmuland

J. W. Mark

Date: Oct. 2/92

Abstract

A flexible bandwidth call service is proposed in this thesis. Its call control protocol is designed and specified. The new protocol is based on a new admission control and bandwidth allocation scheme, namely the *flexible access control technique* proposed in this thesis. Impact of the new service, protocol and admission control and bandwidth allocation scheme on network performance is studied through theoretical analysis and simulation. Practical implementation issues are also discussed.

This thesis proposes a flexible bandwidth call service concept and its call control protocol. Bandwidth demands are flexible and bandwidth allocations can be changed dynamically for calls in-progress. A new access control scheme is designed in the thesis. The new scheme allows bandwidth to be relinquished from and re-allocated to in-progress calls. Bandwidth allocated to each call can be reduced through relinquishment when the traffic load is high, and increased through re-allocation when the traffic load is low. This thesis presents an extension of the conventional knapsack problem to the flexible knapsack one and its solution algorithms. The flexible knapsack problem is used to model the flexible access control in order to maximize instantaneous rewards. In addition to performance analysis of the new protocol and access control scheme, the existence of a threshold structure for optimal dynamic access control policies is proved in the thesis. Protocol validation and simulation studies are also reported. Discussions on practical applications of the results is presented in this thesis.

This study is of immediate benefit to both network designers and service providers.

Acknowledgements

First, I am most indebted to my family, my wife Yian Leng, my brother and parents, I would not be able to complete this thesis without their love and support. I hope I can do the same for them to keep my parents healthy and happy, and to see Yian Leng through her PhD program and start building her dream home soon.

I am also most grateful to my supervisor, Professor Ursual M. Maydell, the most responsible and approachable professor I have ever known, for her support, guidance and friendship.

I have also benefited from Dr. Michael H. MacGregor, my "thesis brother", from the inspiration, who has always been ready to offer help since the very first day we met.

I also benefited a great deal from my supervisory committee members, Dr. Ahemd Kamal for very detailed comments, Dr. Pawel Gburzyński for the wonderful SMURPH and patient help, and Dr. Byron Schmuland for the constructive comments on an early draft and the proof of a theorm. Without their continuous directions and encouragement during the past two years, this thesis would not be finished on schedule. My thanks also goes to the external examiner, Professor Jon Mark, for his thorough reading of the thesis and comments.

Contents

1 Introduction 1

1.1 Overview . . . 1

1.2 Thesis Scope and Objectives . . . 2

1.3 Organization . . . 2

2 Related Work and Motivations 4

2.1 Future Broadband Communication Networks . . . 4

2.1.1 Relevance of B-ISDN to Computer Communications . . . 4

2.1.2 B-ISDN Architecture and Protocol . . . 5

2.1.3 B-ISDN Signaling . . . 10

2.2 Access Control in Broadband Networks . . . 10

2.2.1 Congestion in Broadband Networks . . . 11

2.2.2 Why Congestion Problem Is Still Important . . . 11

2.2.3 Congestion Control Objectives . . . 12

2.2.4 Classification of Congestion Control in B-ISDN . . . 13

2.3 Access Control Policies . . . 15

2.3.1 Reactive Admission Control . . . 16

2.3.2 Optimal Preventive Admission Control Policy Structure . . . 16

2.3.3 Reactive Bandwidth Allocation . . . 17

2.3.4 Preventive Policies . . . 18

2.4 Summary and Interesting Problems . . . 22

2.4.1 Reactive Control Versus Preventive Control . . . 23

2.4.2 Call Setup Queue . . . 23

2.4.3 Negotiable Bandwidth . . . 24

3 Flexible Bandwidth Service and Protocol 25

3.1 Feasibility of Flexible Bandwidth Service . . . 25

3.1.1 Traffic Controllability: User's Prospective . . . 26

3.1.2 Traffic Controllability: Equipment Capability . . . 26
3.2 Possible Benefits of Flexible Bandwidth Service . . . 27
3.3 Flexible Bandwidth Service Definition . . . 30
3.3.1 Service Primitives . . . 30
3.3.2 Service Primitive Parameters . . . 31
3.3.3 Time Sequence of the Primitives . . . 32
3.3.4 Services Required . . . 33
3.4 Protocol Specification . . . 34
3.4.1 Protocol Data Units . . . 34
3.4.2 Signaling Protocol Details . . . 35
3.5 Compatibility with Existing Protocols . . . 38

4 FACT: the Flexible Access Control Technique **40**
4.1 The Basic Policy . . . 40
4.1.1 Application of FACT . . . 41
4.1.2 Effective Control Region . . . 42
4.1.3 Classification of FACT . . . 43
4.2 Variants of the Basic Policy . . . 44
4.2.1 Even and Uneven Relinquishment Policies . . . 45
4.2.2 Immediate, Delayed and Early Relinquishment . . . 46

5 Static Allocation **49**
5.1 Conventional Knapsack Problems . . . 50
5.2 The Flexible Knapsack Problem . . . 51
5.2.1 Flexible Knapsack Problem Definition . . . 51
5.2.2 Practical Applications of the Flexible Knapsack . . . 52
5.2.3 Flexible Knapsack Versus Conventional Knapsack . . . 52
5.2.4 Modeling the Flexible Access Control Problem . . . 54
5.3 Solutions to Flexible Knapsack Problems . . . 55
5.3.1 Solution to the Unconstrained Flexible Knapsack Problem . . . 55
5.3.2 Solution to Constrained Flexible Knapsack Problems . . . 55
5.3.3 Heuristic Solution to Constrained Flexible Knapsack Problems . . . 63
5.4 Numerical Results . . . 64
5.4.1 FACT Versus Conventional Scheme . . . 64
5.4.2 Exact Solution Versus Heuristic Solution . . . 66
5.5 Summary . . . 67

6 Dynamic Allocation: Analytical Performance Study **70**

6.1 Overview . . . 70

6.2 General Assumptions . . . 71

6.2.1 Measurement Units . . . 71

6.2.2 Network Activities . . . 71

6.2.3 Traffic Model Assumptions . . . 72

6.3 Comparable Fixed Allocation Models . . . 75

6.4 Single Flexible Class at the VC Level . . . 76

6.4.1 Assumptions and Model . . . 77

6.4.2 Performance Measures and Comparisons . . . 81

6.5 Single Flexible Class at the Burst Level . . . 94

6.5.1 Assumptions and Model . . . 94

6.5.2 Performance Measures and Comparisons . . . 95

6.6 Mixed Traffic of Fixed and Flexible Calls . . . 99

6.6.1 Analytical Model . . . 100

6.6.2 Performance Measures and Comparisons . . . 102

6.7 Multiple Classes . . . 105

6.8 Summary . . . 107

7 Dynamic Access Control: Optimal Policies **124**

7.1 Overview . . . 124

7.2 Model for Mixed Classes of Fixed and Flexible Calls . . . 125

7.3 Optimal Policies Structure . . . 126

7.3.1 Coordinate Convex Policies . . . 126

7.3.2 Reward Function . . . 127

7.3.3 Threshold Type Policies . . . 131

7.3.4 Optimal Policy Structures . . . 133

7.4 Discussions of Multiple Classes . . . 136

8 Simulation Study **138**

8.1 Overview . . . 138

8.2 Simulator Configuration . . . 140

8.2.1 Traffic and Network Nodes . . . 140

8.2.2 Signaling Protocol . . . 143

8.2.3 Conditions and Assumptions . . . 144

8.3 Control Parameters . . . 145

8.4 Processing of Captured Events . . . 146

8.5 Simulator Verification and Validation 150
8.6 Protocol Validation Through Simulation 151

9 Results and Practical Application **154**
9.1 Network Particulars 154
9.2 An Application in Network Design 155
9.2.1 Bandwidth Provisioning 156
9.2.2 Emergency Handling 160
9.2.3 Signaling Capacity 160
9.3 An Application in Network Operation 162
9.3.1 Benefits of Upgrading to FACT 162
9.3.2 Optimal Blocking 162
9.4 Protocol Validation Through Simulation 163

10 Conclusion **164**
10.1 Summary of Research 164
10.2 Contributions and Their Novelty 166
10.3 Directions for Future Research 166

List of Tables

2.1 Layered Model for Admission Control 14

3.2 New Service Parameters for Flexible Call Class 31

5.3 Cases for Obtaining the Optimum 59
5.4 Comparison of Average Total Reward: FACT vs. Conventional 65
5.5 Average Utilization Comparison: Exact vs. Heuristic Solutions 68

6.6 Summary of Notations for Single Class Call Study 79

8.7 Example of Event Log 146
8.8 Validation Techniques Considered and Applied 151

9.9 Blocking Probability (%): FACT versus Fixed Allocation 157
9.10 Simulation Results: Number of In-Progress Calls 159
9.11 Simulation Results: Call Blocking Rates 160
9.12 Signaling Load Investigation 161

List of Figures

2.1 Virtual Path and Virtual Channel Structure 6
2.2 B-ISDN Transport Connection Model 6
2.3 B-ISDN Architecture 7
2.4 Broadband Protocol Model 8
2.5 SONET Frame With Flexible VT Mapping 9

3.6 Example Network: Alberta Medical Network 28
3.7 Time Sequence Diagram for Flexible Bandwidth Service at the Network Layer 33
3.8 Flexible Call Control Protocol at User Side of UNI 36
3.9 Flexible Call Control Protocol at Network Side of UNI 37

4.10 Effective Control Region for Flexible Calls 43
4.11 When to Relinquish: Immediate, Delayed or Early 47

5.12 Example of a Flexible Knapsack Problem with Two Subclasses 53
5.13 Relation Between Flexible and Conventional Knapsacks 54
5.14 Average Computation Time of Exact vs. Heuristic Solutions 67

6.15 Traffic Model for Single Class of Flexible Calls/Bursts 73
6.16 Access Control Model for Single Class of Flexible Calls 74
6.17 Holding Time Distribution 74
6.18 State Diagram for Single Class of Flexible Calls 80
6.19 Probability Distributions Versus Call Numbers for Single Class of Flexible Calls . . . 109
6.20 Lowest Blocking Among All Fixed Allocation Policies 110
6.21 Lowest Blocking Policy Among Those Serving Maximal Number of Calls 110
6.22 Call Blocking Probability Comparison: FACT vs. Minimum Allocation 111
6.23 Queueing Probabilities For Single Flexible Call Class 112
6.24 Queueing Delay Comparison for Single Flexible Call Class 113
6.25 Call Delay for Single Class of Flexible Calls 114

6.26 Relinquish Probabilities for Single Class of Flexible Calls 115
6.27 State Diagram for a Single Class of Flexible Bursts 115
6.28 Model for Mixed Traffic 116
6.29 State Transitions for Mixed Calls 116
6.30 Blocking Rates for Mixed Calls with Constant m 117
6.31 Blocking Rates for Mixed Calls with Variable m 118
6.32 Call Delay for Mixed Traffic with Fixed m 119
6.33 Call Delay for Mixed Traffic with Variable m 120
6.34 Bandwidth Utilization for Mixed Traffic with Variable m 121
6.35 Relinquishment Probability for Mixed Traffic with Variable m 122
6.36 State Diagram for Two Flexible Classes 123

7.37 Unified Model for the Call and Burst Levels with Mixed Traffic 125
7.38 Subset Reward Less Than Full Set Reward 129
7.39 Threshold Policies 131
7.40 Corner Points 132
7.41 Position of Type-2 Corner Point 134
7.42 Irreversibility Example for Multiple Flexible Classes 137

8.43 Simulation Environment 139
8.44 Network Model for Simulation 141
8.45 Simulator Topology 142
8.46 Bandwidth Allocation During Load-Up and Settle Phases 148
8.47 First Blocking Indicates the End of Transient Period 149

9.48 Identifying Equilibrium Phase for STS-84 through Number of Calls 158
9.49 Simulation Results: Bandwidth Allocations 159
9.50 Capacity Available to Emergency Demands 161

List of Symbols

C	VP bandwidth capacity,
F	Flexible knapsack capacity
L	$= C/b_{max}$ the minimal number of calls to fully utilize the VP,
H	$= C/b_{min}$ the maximal number of calls to fully utilize the VP,
N_p	number of calls in progress
N_q	number of calls in the queue
k	$= N_p + N_q$, state of the birth-death process, number of calls in the network
$\lambda(k)$	mean arrival rate when the state is k,
$1/\mu$	mean amount of information,
μ_{min}	$= \mu b_{min}$, call departure rate when b_{min}, instead of 1, BUs are used
μ_{max}	$= \mu b_{max}$, call departure rate when b_{max}, instead of 1, BUs are used,
μ_k	state transition rate from k to $k-1$,
ρ	$= \lambda/\mu$,
ρ_{min}	$= \lambda/\mu_{min}$,
ρ_{max}	$= \lambda/\mu_{max}$,
u	$= \rho/C = \rho_{max}/L = \rho_{min}/H$,
u_{min}	$= \rho_{min}/C$,
u_{max}	$= \rho_{max}/C$,
S	queueing threshold before relinquishment
h	$= H + S - L$ number of states from L to $H + S$
D_p	call holding time, or transmission time
D_q	call queueing time
D	$= D_p + D_q$, call delay, the time spent in the network
p_k	equilibrium probability of system being in state k
P_k	$= \sum_{i=0}^{k} p_i$, the equilibrium probability of that system being in state $j \leq k$
m	limit on the number of fixed calls under high load of flexible calls
mix	ratio of flexible calls to all fixed and flexible calls
n_1	number of class-1 (fixed) class in the network
n_2	number of class-2 (flexible) class in the network
$\mathbf{n}$	(n_1, n_2)
$\mathbf{n}_1^-$	$(n_1 - 1, n_2)$
$\mathbf{n}_1^+$	$(n_1 + 1, n_2)$
$\mathbf{n}_2^-$	$(n_1, n_2 - 1)$
$\mathbf{n}_2^+$	$(n_1, n_2 + 1)$
b	bandwidth allocation to class-2 (flexible) calls
Ω	state set for a CC access control policy, also the policy itself
Ω_F	set of all possible states
$\mathbf{b}$	column vector for bandwidth allocation for both classes
$\mathbf{r}$	column vector for reward rates for both classes
$R(\Omega)$	reward function for policy Ω.

Chapter 1

Introduction

1.1 Overview

The users of communication networks expect the network to meet certain minimal performance requirements. Performance studies have concentrated on how to provide adequate capacity to achieve or exceed users' performance requirements. Network operators [1] are always willing to provide service required by users for a fee. Realistically, users' affordability is limited. Thus, communication demand is actually constrained between the minimal performance requirement and maximal affordability. Excessive capacity provisioning could be as harmful to network operators as inadequate capacity provisioning. Because both over-provisioning and under-provisioning have ultimately the same consequence, that is financial insolvency of the network operators in a competitive market.

Bandwidth demand for many communication applications is in fact a natural range limited by the minimal performance requirements at one end, and the affordability or user premise equipment capacity at the other. Based on this observation, this thesis proposes a new service class for broadband integrated services digital network (B-ISDN) based on its traffic negotiation and re-negotiation capability. This new service allows users (calls) to specify a range for their acceptable bandwidth demand. Any bandwidth within this range can be assigned at the call setup stage, and changed dynamically for the duration of the call. This service is referred to as flexible bandwidth service. For this new service a new admission control and bandwidth allocation scheme is also proposed to achieve low blocking under high network load, and high utilization under light network load. This scheme is based on the (re-)negotiation capability of B-ISDN. It has been recognized within the International Telegraph and Telephone Consultative Committee (CCITT) that "the traffic negotiation capability at connection establishment and re-negotiation capability of an already established connection are typical requirements in B-ISDN signaling" [37]. However, the associated signaling

[1] The notion "network operator" is used here to refer to the communication network management and service provider.

protocol has not been designed yet, nor has the performance impact of (re-)negotiation been studied. The objective here is to address these two open problems by formalizing a bandwidth negotiation and re-negotiation and analyzing the associated performance impact. Performance of the new service scheme is analyzed and compared to the conventional service schemes. Simulation is used to complement the analytical study. This thesis also conducts protocol implementation design of the new scheme and discusses the performance study results in a practical scenario.

1.2 Thesis Scope and Objectives

Admission control and bandwidth allocation are dealt with at the call level for broadband communication networks. "Broadband communication networks" are considered to be based on fast-packet switches and outband signaling techniques and to provide integrated connection-oriented digital services, such as B-ISDN with the asynchronous transfer mode (ATM) technique. Admission control is defined by CCITT as "the procedure within the control part of network nodes used to decide whether or not a request for a (virtual) connection can be accepted based on the requested usage parameters and already established connections" [8]. Bandwidth allocation decides how much bandwidth should be allocated to each accepted connection. Admission control and bandwidth allocation are also collectively referred to as access control, because they together determine if and how connections access the network bandwidth service. For terminology clarification, the two terms, "access control" and "admission control and bandwidth allocation" are used interchangeably.

The main objectives of the thesis are the following:

1. To design a flexible call service which takes advantage of the fact that communication demand is a range rather than a fixed value,
2. To define a call control protocol that provides the flexible bandwidth service,
3. To propose a new access control policy which is to balance both the network operators' and subscribers' interests,
4. To study the performance impact of the flexible bandwidth service using both theoretical analysis and simulation, and
5. To investigate corresponding implementation and practical application issues.

1.3 Organization

This thesis is divided into ten chapters. Chapter 2 presents related work in the area of admission control and bandwidth allocation for broadband networks. The new flexible bandwidth service and

the call control protocol are motivated and defined in Chapter 3. That chapter also contains the service definition and protocol definition documents. This is followed by a chapter describing the new admission control and bandwidth allocation technique designed for the new flexible bandwidth service.

When future call arrivals and departures are unknown or not considered, the access control is a static one dealing with instantaneous traffic. This problem is studied in Chapter 5, where both exact and heuristic algorithms for the flexible knapsack problem are obtained.

The dynamic bandwidth allocation problem with stochastic call arrivals and departures is studied in Chapter 6, and the optimal dynamic admission control problem is solved in Chapter 7.

Following the theoretical analyses in Chapters 5, 6 and 7, a generic network simulator for investigating network performance under more general or more realistic conditions is presented in Chapter 8. Then, in Chapter 9, the results are interpreted through some practical network design and operation situations. Finally, Chapter 10 concludes with a summary of results and their novelty, and a discussion of open problems for future studies.

Chapter 2

Related Work and Motivations

In this chapter, the architecture and protocol of the broadband network under consideration are first summarized. Next, general congestion control issues are discussed. This is then followed by an overview of related work on admission control and bandwidth allocation. This chapter is concluded by a discussion of interesting open problems to provide motivations for the study presented in the thesis.

2.1 Future Broadband Communication Networks

Broadband is defined as "a service or system requiring transmission channels capable of supporting rates greater than the primary rate" at 1.5 Mb/s [8]. The future broadband communication network considered in this thesis is the broadband integrated services digital network (B-ISDN) . This section addresses the following two questions: Why is B-ISDN relevant to computer communication and what are the characteristics of B-ISDN?

2.1.1 Relevance of B-ISDN to Computer Communications

The future communication and telecommunication networks will evolve eventually into a single network providing a unified broadband access and switching methods for voice, data, video and other digital streams; that network is B-ISDN.

It has been commonly agreed upon that computers will be communicating within metropolitan areas or farther using B-ISDN as a "digital highway." Computer communications will be a major service provided by B-ISDN. For instance, it is expected that inter-LAN traffic will account for up to half of all switched high speed data traffic by 1995 [67]. Bellcore recognized "LAN interconnection is a primary driver for B-ISDN development" [70]. Northern Telecom also concluded that the

interconnection of LAN's and MAN's using the synchronous optical network is "the most promising early opportunity for broadband services" [67]. All these indicate the magnitude of reliance of computer communications on B-ISDN.

With the advent of information and communication services, the scope of computer communications will also widen to cover areas which were traditionally considered as "non-computer" areas. Many non-computer applications serviced by B-ISDN will be indistinguishable from "pure" computer communications. For example, as far as communication is concerned, there is little difference between a remote file server and the future individualized newspapers and manipulatable movies [45], or between multi-database access and video browsing [6]. All these applications can be considered as computer communications, and then are subject to the same traffic control and network management policies.

2.1.2 B-ISDN Architecture and Protocol

B-ISDN is based on fiber optic transmission. The international standard for interfaces based on fiber transmission is the synchronous digital hierarchy [1],[3]. It is also known as the Synchronous Optical Network (SONET) in North America [70],[4]. This thesis uses the terminology SONET. The transport and switching technique for B-ISDN is defined by CCITT to be the Asynchronous Transfer Mode (ATM) [9].

This subsection provides a brief overview of the B-ISDN architecture and protocol. It shows that B-ISDN has a layered structure at the virtual channel, the virtual path and the transmission levels. This leads to the layered access control model presented in the next section. The similarity between different network levels is also discussed, which leads to the application of the same access control technique at each level. Furthermore, it shows that B-ISDN supports flexible bandwidth provisioning at the lower layer level, and there is a large amount of signaling capacity to support new network control activities.

ATM Concept

The most important features of ATM are the fixed length cell and the cell header label. Information is transferred as a 53-octet unit called "cell." The cell header label of five octets contains a virtual channel identifier (VCI) and a virtual path identifier (VPI). Virtual path (VP) is a collection of virtual channels (VCs) with the same VPI as a switching and transport entity. Figure 2.1 shows the VP and VC bundling in a physical channel.

The VCI in an ATM cell is used to distinguish VC links inside a VP. VC links and VP links are, respectively, established or terminated by the assignment or removal of VCI and VPI values.

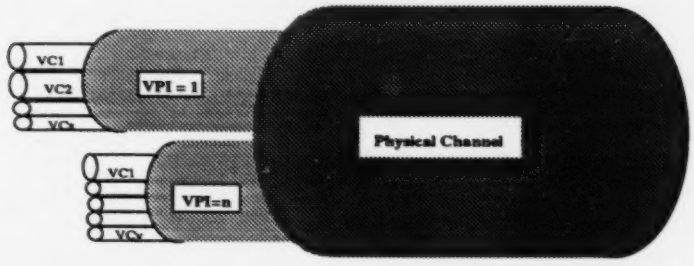

Figure 2.1: Virtual Path and Virtual Channel Structure

VC links and VP links are catenated to form a VC connection (VCC) and a VP connection (VPC), respectively. Figure 2.2 shows the connection model for B-ISDN.

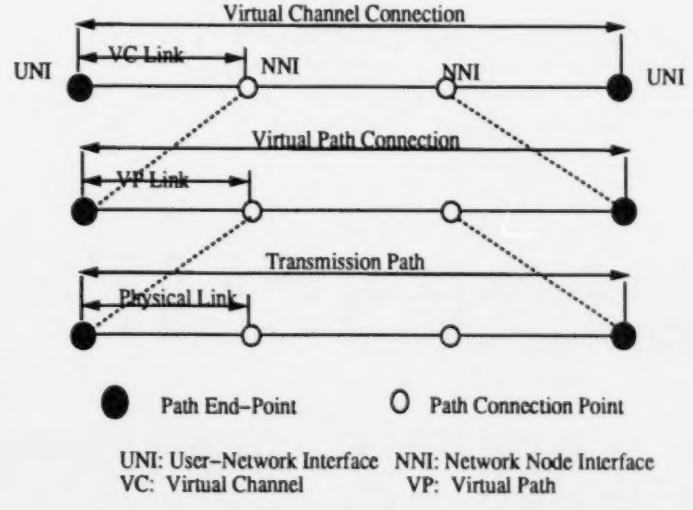

Figure 2.2: B-ISDN Transport Connection Model

B-ISDN Architecture

The B-ISDN architecture itself can also be viewed according to these three levels: transmission level network, VP level network, and VC level network [37]. Figure 2.3 illustrates the architecture of B-ISDN in this layered view. Each level manages transport entities of local interest only. For instance, the VP level network is concerned with VP management only. The traffic of individual VCs inside a VP is not regulated by the network, only the aggregate traffic of all VCs within a VP

is managed by the network [71].

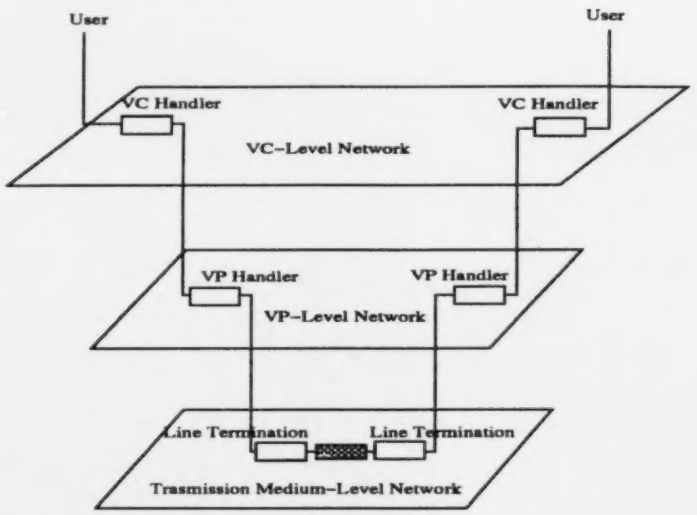

Figure 2.3: B-ISDN Architecture

B-ISDN Protocol

Figure 2.4 shows the low layers of the protocol stack for ATM-based B-ISDN. Admission control and bandwidth allocation usually are considered as the functionality of network and/or transport layers of the ISO reference model for open system interconnectivity (OSI) [37].

SONET Overview

While the format for a *sonnet* is 14 lines of variable length, the basic format for SONET is, however, 9 lines (rows) of fixed length at 90 octets per line. It is called the Synchronous Transport Signal - Level 1 (STS-1), and is the first level of the interface family and the basic building block for SONET frames. High rate SONET signals are obtained by byte-interleaving N frame-aligned STS-1's to form an STS-N. N STS-1's can also be concatanated to form an STS-Nc. Each level-N frame is transmitted every 125 μs, supporting the 1 byte/125 μs = 64 kb/s channel at each byte position. An STS-1 thus has an aggregate capacity of 51.84 Mb/s.

The term "overhead" is used for the provision of network management information, while "payload" indicates SONET users' digital information.

The optical carrier level 1 (OC-1) is obtained from STS-1 after scrambling (to avoid long strings

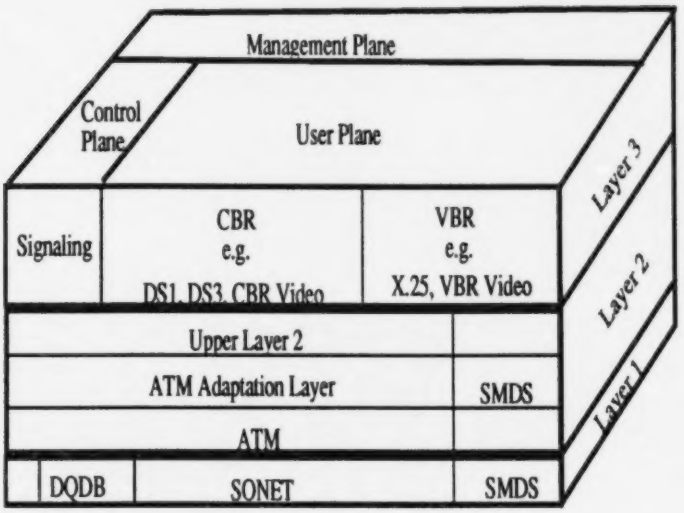

Figure 2.4: Broadband Protocol Model

of ones and zeros and allowing clock recovery at the receiver) and electrical-to-optical conversion. The OC-1 is the lowest level optical signal to be used at SONET equipment and network interfaces [4].

SONET Flexible Bandwidth Provisioning

Flexible bandwidth provisioning is supported by SONET at different levels.

A pointer action byte exists in the line overhead of STS-1 for adjusting payload for extra bandwidth or unneeded bandwidth. But it is supposed to be used at most once in every four frames. More substantial bandwidth fluctuations can be handled by payload mapping at the STS level or sub-STS level.

At the STS level, different payload configuration options are possible. For instance, a 622.08 Mb/s OC-12 can either carry one STS-3c with nine STS-1 as an STS-12 or four STS-3c as an STS-12c. Bandwidth provisioning can be changed dynamically from the first configuration to the second one.

At the sub-rate virtual tributary(VT) level, one STS-1 can be comprised of either 7 VT6 at 6.912 Mb/s (DS2) each, or 4 VT1.5 at 1.728 Mb/s (DS1) with 3 VT2 at 2.304Mb/s plus 2 VT3 at 3.456 Mb/s (DS1c) and 1 VT6 at 6.912 Mb/s (DS2). Figure 2.5 depicts three consecutive STS frames with three different VT mappings. It can be imagined, for instance, that the VTs in the

middle of each of the first two STS frames give up bandwidth for the new VT in the next frame so that the VT number increases as the VTs reduce their bandwidth.

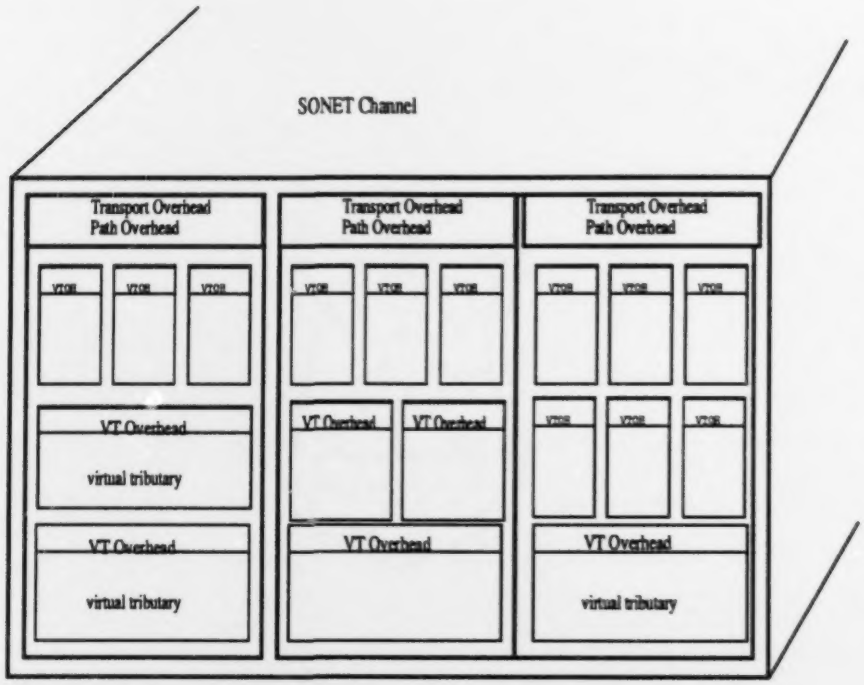

Figure 2.5: SONET Frame With Flexible VT Mapping

Furthermore, individual channels can also be added or dropped dynamically with SONET, which is known as flexible channel provisioning of SONET.

SONET Signaling Capability

Each STS-1 has three types of overhead: path overhead, section overhead and line overhead. The division of overhead types reflects the segregation of processing functions in network elements (equipment) and prompts understanding of the overhead functions.

The section layer contains those overhead channels that are processed by all SONET equipment including re-generators. Apart from management for network maintenance and operator applications, there is a section data communication channel designated for message signaling for network management. It has the capacity of 192 Kb/s.

The line overhead is processed at all SONET equipment except re-generators. The management

capacity for the line layer is nine bytes per frame, or 576 Kb/s for STS-1.

The path overhead is processed at the SONET STS-1 payload terminating equipment; that is, the path overhead is part of the SONET STS-1 payload and travels with it. STS-1 path overhead bytes of interest to bandwidth allocation are the signal label to identify the payload type, the path status to carry maintenance signals, the user channel and three bytes for future growth.

Signaling capability is also provided at the sub-rate VT level by byte V5 of the VT path overhead.

In summary, SONET is capable of supporting flexible bandwidth allocation at all levels including the STS and the sub-rate VT levels. It also provides adequate signaling transport capacity for future network management tasks. The type of signaling protocols that are defined to use all these signaling capacities is the issue of the next subsection.

2.1.3 B-ISDN Signaling

B-ISDN signaling protocols are still in the development phase at CCITT [37]. Presently, as the first step, CCITT is considering the B-ISDN signaling protocol at the user access point based on the current ISDN protocol Q.931 [7], and the signaling protocol between network nodes based on the ISDN User Part of the current Common Channel Signaling System No. 7.

In addition to current ISDN standards, B-ISDN signaling will also be able to control ATM virtual channels and virtual paths, including establishment, release and reconfiguration of VCs and VPs on demand, and on a semipermanent or permanent basis.

It has been recognized within CCITT that "the traffic negotiation capability at connection establishment and re-negotiation capability of an already established connection are typical requirements in B-ISDN signaling" [37]. The signaling protocol for negotiation and re-negotiation has not been designed yet, nor has the performance impact of (re-)negotiation been studied. It is the main objective of this thesis to address these two open problems.

The remaining part of this chapter provides an overview to the access control in B-ISDN, and then discusses in more detail the motivations for the new study.

2.2 Access Control in Broadband Networks

Access control is part of the congestion control function in broadband networks. The other part is traffic enforcement which ensures that in-progress calls do not use more than allocated bandwidth [10]. This section starts with a brief discussion on the need for congestion control in B-ISDN, and goes on to what a good policy should be, before a summary of related work is presented.

2.2.1 Congestion in Broadband Networks

For any time interval, if the total sum of demands on a resource is more than its available capacity, the resource is said to be congested during that interval. In a communication network, congestion occurs when the aggregate instantaneous traffic demands of all in-progress calls exceed the capacity of the resource.

The resource in a communication network may be link bandwidth, buffer space or processing (switching) capacity. When faster links, larger buffers and faster processors are employed, one may naturally expect congestion to disappear, or at least to become insignificant. It is, for example, stated by Tanenbaum [66] that

> "the problem [of congestion] boils down to not enough IMP [interface message processor] buffers. Given an infinite supply of buffers, the IMP can always smooth over any temporary bottlenecks by just hanging onto all packets for as long as necessary."

But this is not always the case. Congestion is still an important challenge in B-ISDN as explained in the next subsection[35].

2.2.2 Why Congestion Problem Is Still Important

Congestion control is still important in broadband networks, even though the technical progress provides large buffers, fast communication links and fast processors affordable. The reasons can be given as follows.

1. The congestion problem cannot be solved with a large, or even infinite, buffer space as shown in previous studies[53], because the queue and delays can get so long that by the time the packets come out of the switch, most of them have already timed out and have been retransmitted by higher layers. In fact, too much buffer space is more harmful than too little buffer space, since the packets have to be dropped after they have consumed precious network resources.

2. The congestion problem cannot be solved with high-speed links either. The high-speed links cannot exist in isolation, and the low-speed links do not disappear as the high-speed links are added to a network. B-ISDN has to inter-network with existing low speed networks. Packet arrival rates may become much higher than departure rates at some nodes. Hence, congestion is incurred by the accumulating packets [33].

3. Introduction of high-speed processors or switches cannot solve the congestion problem either, owing to the increase of the chances of congestion and possible speed mismatch [35].

Moreover, congestion control is still needed in high-speed networks during those instances of resource overloads due to unforeseen traffic focusing and peaking, as well as during network component failures. High transmission rates allow little time for the network to react to such overloads before severe congestion occurs.

Congestion control is considered as one of the fundamental challenges facing broadband networks. It is believed that congestion is a dynamic problem, and cannot be solved with static solutions alone [34]. The solution requires effective combinations of different types of policies such as rate control, preventive admission and reactive bandwidth allocation.

2.2.3 Congestion Control Objectives

The objectives of congestion control are to establish fair blocking and to ensure that the resources available to each admitted connection are sufficient to meet performance objectives for transfer delay and reliability.

If a traffic stream is of a constant rate for the duration of the call, it is referred to as constant bit rate (CBR) traffic. A CBR call can be accepted if the bandwidth available on each link along its route exceeds its bandwidth demand. Considering the stochastic arrival and departure of calls, sophisticated controls can be applied to optimize the network performance over long runs. Although there is enough bandwidth available, a call may still be blocked under optimal blocking policies to preserve bandwidth for the possible arrival of some other calls that are more beneficiary than the former call [60]. Some previous studies [58] [41] have mentioned that CBR traffic with negotiable bandwidth usage (transmission rate) can be supported. They, however, all lack a negotiation protocol and the performance analysis on negotiation.

Some traffic streams in broadband integrated networks are bursty in nature. This traffic is referred to as variable bit rate (VBR) traffic. VBR traffic streams can be of either the on-and-off type or the continuous type. For on-and-off type of bursty traffic, each burst can be viewed as a "mini-call" of CBR, thus the same access control policy at the call level applies. For continuous type VBR traffic, admission control is based on peak rate, peak duration and other estimates of traffic characteristics. Rate control then must be used to ensure that the transmission rate agreed upon at admission time is indeed observed by the users.

Now, in summary, a good congestion control scheme should have the following attributes.

- Efficiency. The scheme must have low overhead. In particular it should not increase data traffic during congestion. There is a fine difference between applying it to computer networks and telecommunication networks. Traditionally, it has been considered impractical to increase control (signaling) messages when the load is high in a computer network, as network control

information and data traffic share the same link, and thus contend for the bandwidth. Integrated digital networks, however, use separate channels for signaling and data. Congestion in the data network does not imply congestion in the signaling network. Thus, it is feasible to increase the signaling load when the data network is congested or approaching congestion.

- Fairness. All calls of the same class should be treated equally, arbitrary call blocking or degradation should be avoided. At the connection level, it is necessary that the usage of high-bandwidth services not consume available resources to such an extent that lower bandwidth services experience an uncontrolled increase in blocking probability. This is the concept of fair blocking.
- Responsiveness. The scheme must be responsive and match the demand dynamically to the available capacity.
- Robustness. The scheme must work in bad environments such as unexpected traffic surge.
- Global Optimization. The scheme must be socially optimal. That is, the scheme must allow the total network performance to be maximized.
- Effectiveness. The scheme must be able to control congestion.

2.2.4 Classification of Congestion Control in B-ISDN

Congestion control schemes can be classified based on either resource management or the responsive nature of the policy. Different types of congestion control policies can then be applied to various levels of the network.

Resource Management Classification

Congestion control schemes can be classified into two types based on resource management: resource creation schemes and demand reduction schemes.

Resource creation schemes increase the capacity of the resources by dynamically reconfiguring resources. The network is solely responsible for solving the congestion problem, and users do not need to be informed. For example, this thesis shall consider resource creation at the VP level. VP capacity can be increased whenever necessary and possible, so that the calls can be accepted at the VC level transparently. The previous section has shown that flexible reconfiguration of payload capacity can be supported by SONET, say, at VT or STS levels.

Demand reduction schemes include service denial, service degradation, and scheduling schemes. This thesis is also motivated by the fact that communication users may consider service degradation more preferable than service denial.

Preventive and Reactive Control

Another classification of congestion control schemes could be based on whether a scheme is reactive or preventive. Reactive control is to relieve congestion once it has occurred in some part of the network. Preventive control, on the other hand, prevents the occurrence of congestion. A good policy should be a combination of both preventive control and reactive control. The policy should prevent long term congestion, and react to short term congestion. Congestion prevention is normally achieved by resource allocation (reservation), while reactive control is often based on restricting resource usage during congestion.

Window control is an example of reactive control schemes. A typical example of preventive control is access control which decides whether or not the new request can be accepted. The simplest access control strategy is peak rate allocation. This allocates the required maximum bandwidth to every request regardless of other characteristics of traffic, or rejects the request if the sum of all allocated bandwidth values exceeds a certain threshold. The threshold level is defined by the bandwidth available in the network and by the quality of service requirements of the new request. The objectives of access control are to establish fair blocking and to ensure that the resources available to each admitted connection are sufficient to meet performance objectives.

Layered Control

As shown in Figure 2.3, B-ISDN has a layered architecture. Congestion may occur at any of the network layers. Hence, another classification of congestion control policies can be based on the network layers. Although there is no standard divestiture of congestion control levels, most models are very similar. The layered model considered in this thesis contains most existing models.

Table 2.1: Layered Model for Admission Control

Level	Cooper's [14]	Gallassi's [20]	Hui's [31]	Hong's [29]
VP	–	network	–	–
VC	connection	call	call/3	call
Burst	packet	–	burst/2	–
Cell	ATM	cell	packet/1	cell

Four levels are present in this model: virtual path, virtual channel, burst and cell levels. Table 2.1 tabulates the correspondence of this model to some existing models. The VP and VC level controls are to avoid long term congestion and maintain the traffic load at a manageable level, whereas the burst and cell levels are to avoid and resolve short term congestion. The higher level controls are exercised less frequently than the lower levels, and hence, they are more preventive and less responsive. The burst-level admission control and bandwidth allocation is necessary especially

Spooled Output

For: Shuang Deng

UID: shuang@armena

Pages: 14

JobName: stdin

Date: Fri Oct 2 23:23:05 1992

Printer: LW654a:LaserWriter@*

for image retrieval or very-high-speed data services.

The responsibilities of each level are the following:

- VP Level: to allocate dynamically VP capacities to achieve low network blocking and high bandwidth utilization
- VC Level: to control VC access and administrate VC bandwidth allocation for optimal call blocking in the VP.
- Burst Level: to decide acceptance or denial of bursts without incurring unacceptable burst blocking
- Cell Level: to ensure cell loss, delay and delay variation within the requirements, to monitor a connection's resource usage for compliance with appropriate limits and to act on violations of those limits.

Usually it is insufficient to have congestion control at one level only, or of one type only. An effective scheme should be a combination of preventive and reactive types. For instance, preventive control at the VC level could be used at call setup time to ensure sufficient resource availablity to meet the requirement of all calls. For the duration of the calls, at the cell level, a traffic enforcement policy is exercised at the network access point, and a buffer management policy is used at the switching node inside the network to prevent congestion and to ease congestion. The burst level control is used to set the parameter for cell level traffic enforcement such as the token pool size if the cell level uses the leaky bucket policy [69]. This thesis is interested in admission control and bandwidth allocation, which is applicable to the first three levels only, namely VP, VC and burst levels. It is assumed that some kind of traffic enforcement policies, such as leaky bucket and its variants [69] [13] [64] [2], is used at the cell level to ensure that bandwidth allocation is observed at the cell level as well as to carry out the other cell management functionalities.

The remainder of this chapter provides a brief overview of existing admission control and bandwidth allocation policies.

2.3 Access Control Policies

Access control contains two inseparable parts: controlling call admissions to the network and managing bandwidth allocation among calls that have been granted network access, i.e., admission control and bandwidth allocation.

Admission control is exercised at each link of the route for a connection. It decides whether the connection request can be accepted or not on the link. Acceptance will be granted and bandwidth

allocated at the link only if the quality of service estimate is still within an acceptable level after the requested bandwidth is allocated to the new call. A call is allowed to access the network and use the allocated bandwidth if it is accepted by each link on the connection route [59].

Access control excludes the cell/packet level control which is traffic policing or enforcement. For example, leaky bucket [69] and its variants [13] [2] ensure that a call does not exceed its transmission rate limit through using a token pool, whereas access control is responsible for determining the token pool size at the call setup time and, possibly, during the transmission time.

Access controls can be either reactive or preventive. They are discussed in the next two subsections, respectively.

2.3.1 Reactive Admission Control

Reactive admission control can revoke access privileges from in-progress calls. This implies that some calls can be either preempted or suspended when deemed necessary by the network. A preempted call releases channel association with the call, whereas the channel association for a suspended call is kept for later resumption of the communication activities.

This thesis considers call suspension only, as opposed to preemption. The rationale is that the network resources already consumed by a call before preemption would be wasted. In addition, call suspension/resumption procedures are simpler and faster than call preemption/re-setup procedures, because no additional routing and channel association are required. Hence, call suspension is a more promising practical approach than call preemption. To accommodate a high priority or delay sensitive call, the network could just suspend some lower priority or delay insensitive calls and resume them later when bandwidth contention eases. Call suspension amounts to zero bandwidth allocation decisions in a reactive bandwidth allocation scheme. That also simplifies the analytical study later.

2.3.2 Optimal Preventive Admission Control Policy Structure

Preventive admission control attempts to maximize the rewards by optimally rejecting requests even when there are sufficient resources available to accommondate those requests. With the knowledge of stochastic arrival and departure processes, it can "reserve" dynamically resources for future arrivals.

Without the knowledge of the structure, the search for the optimal policy is a time-consuming process, thus limiting its usefulness in broadband networks. The knowledge of the optimal policy structure can greatly reduce the number of candidates. Thus, the search for the optimal policy structure has been an interesting, but difficult, problem in admission control studies.

This optimal admission control problem is not unique to communications, it can be general

resources. Hence, the studies in other areas are applicable to communications, and vice versa. In 1981, Foschini, Gopinath and Hayes first discovered the existence of a certain threshold structure for optimal admission control among two types of customers [19]. Two years later, Foschini and Gopinath further studied this problem for shared memory allocation between multiprocessors [18]. The proof was still limited to two classes. Ross and Tsang, in 1989, further expanded the results from Poisson arrivals to arbitrary arrivals with non-increasing mean interarrival time [60].

All the existing studies consider that customers have fixed resource demands. A new situation will be considered in this thesis. The customer's resource demand will be flexible. Moreover, the usage or allocation of the resource (bandwidth) by a customer can be changed dynamically during the service time. The existence of a certain structure for an optimal policy will then be investigated in this thesis.

2.3.3 Reactive Bandwidth Allocation

Reactive bandwidth allocation schemes adjust bandwidth allocations in response to the network load for the duration of the calls. The scheme relies on a congestion detection mechanism to detect congestion and to initiate the congestion relief mechanism. The traffic fed into the network is reduced (demand reduction) at first, and then increased later. The congestion detection mechanism and congestion relief mechanism are the most important components of reactive bandwidth allocation.

Window Control

Window based policies have been used in computer networks, where each source uses a window to regulate its bandwidth usage or packet transmission rate. The window is the maximal number of packets that the source can feed into network during a certain time interval. Detection of congestion can be accomplished with either a control packet sent directly from the congested node, or an acknowledgment of packet loss from the destination node.

The control packet sent directly from a congested node is called a choking or back-pressure packet. Back-pressure schemes are used in the Signaling System No. 7 [43].

Timeout schemes detect congestion if there is no acknowledgment of receipt within a certain predefined time period. Timeout controls have been used in ISDN to keep track of the number of packets pending in the network for each VC (or aggregation of VCs) [62]. Frame relaying is a packet mode bearer service in narrowband ISDN for data communication [42]. Several window based on flow control schemes were evaluated by simulation [12]. The performance measure used was the "effective layer-3 throughput" or the "goodput" as seen by the users.

However, it seems that the majority consider window control is unsuitable for high speed

networks (e.g., [2]), because the buffer could overflow by high speed traffic before the source is notified of the congestion.

2.3.4 Preventive Policies

A review of various preventive access control policies is provided in this subsection.

Scouts Scheme

The scouts scheme is the simplest and the least expensive preventive scheme, because it does not require any information about the network or other traffic sources such as arrival and departure processes, network capacity, or current occupancy [44]. With the scouts scheme, a set of data packets is sent as "scouts" with traffic characteristics similar to real-traffic at the call setup time. The call is admitted, if the scouts succeed. The main problem is that a false decision could be made because of temporary congestion or idleness in the network.

Numerical Results Guiding Policy

The approach proposed in [20] is to cluster the sources at the cell level into homogeneous groups (classes) characterized by the same traffic descriptors, and then to solve the problem in two steps. First, only homogeneous environments are considered. The relationship between mean offered bandwidth for a specific class and the assigned bandwidth to all classes, is obtained as a curve by simulation, while the cell loss of this class is constrained. Then, the second step is to extend these results to a heterogeneous environment. Assuming the number of in-progress calls is known, one method is to accept a call only if the sum of bandwidths needed to carry the in-progress calls and the new call does not exceed the total link capacity, according to the curve obtained in the first step. This method could be very restrictive when the number of classes is large and the number of calls for each class is small. In this case, another method offers a better solution: the new call is accepted if the resulting total mean traffic could be accepted even if it were offered by the "most demanding" class, i.e., the class with the highest bandwidth requirements. Simulation results are reported in [16].

Random Complete Sharing

Maly *et al.* proposes a distributed algorithm allocating bandwidth dynamically to clusters of nodes [48]. Each node has one channel reserved for it. The remaining channels are kept in a bandwidth pool and allocated on demand in a CSMA/CD manner. It is based on broadband technology and

allows for dynamic allocation of bandwidth in a fair manner.

For homogeneous traffic, the complete sharing policy yields the best throughput [65]. For different classes with heterogeneous traffic, however, complete sharing cannot provide distinctive qualities of service such as different delay and bandwidth requirements to different calls. Another problem is the instability effect on system throughput and user blocking probability as the offered load increases [40].

Bandwidth Partitioning

Accepting a few calls which require high bandwidth has the effect of blocking several smaller calls so that the call blocking probability becomes unfairly high for the smaller calls. To deal with this problem, bandwidth usually is partitioned among different service classes. A connection for a given service class is admitted if and only if there is sufficient bandwidth in the bandwidth pool/segment corresponding to that particular class [14] [26]. The calls of one partition cannot use the bandwidth in the other partitions. The bandwidth partitioning can be implemented as a virtual path.

The next natural step of the complete partitioning concept is movable boundaries as evident in Gerla's and Pazos-Rangel's study on narrowband ISDN[22]. In the context of B-ISDN, it means that the VP sizes can be adjusted dynamically according to the traffic load in each VP [54].

Threshold

Again following the same path in the history of narrowband ISDN, circuit-switched networks or general resource allocation [19] [60] [18], the threshold scheme is the candidate to improving the bandwidth partitioning scheme. The application of threshold control was first considered by Gersht and Lee in 1988 for virtual-circuit control in B-ISDN. They studied a simple threshold-based policy, where the threshold parameters were determined so as to minimize a weighted sum of call blocking probabilities while providing required GOS in terms of blocking performance to each class of calls [23]. Their approach was based on integer programming. A simple approximate solution was given, and its results were validated by simulation.

In 1990, Ilyas and Mouftah suggested that a call should be accepted if its bandwidth requirement does not exceed some percentage of the total available bandwidth [32]. The simulation results show that the arrival rate of high bandwidth calls does not affect the blocking probability of the lower bandwidth calls.

Multilevel Control

Hui proposed a multilevel resource allocation scheme in [31]. There are three network levels: packet, burst and call levels. At the lowest level, a high throughput packet switch is assumed for statistical allocation of the output time slots to packets (cells). Controls are exercised at the burst and call levels. The call level control emulates circuit switching for fixed and high bit rate services and denies calls when facility overload may cause excessive burst blocking. The burst level control emulates fast circuit switching for the individual trunks for avoiding excessive packet blocking within a trunk.

A pilot packet is used to initiate a request for the burst or the call. The pilot packet may contain the stepwise increase of bandwidth at the level of concern. A negative increase indicates the completion of the burst/call, thus releasing that bandwidth. The pilot packet may also contain other parameters of interest, such as an estimate of the holding time for the bandwidth increase.

At the initiation of a call, if the burst blocking probability is acceptable after this call is accepted, then the requested amount of bandwidth is allocated to the call and the chosen routing is recorded. At the arrival of a burst during that call, part of the allocated bandwidth is utilized, the cell blocking probability is calculated, and the burst is admitted only if the cell blocking probability is below the acceptable limit.

The network role in bandwidth allocation is to process passively the request of bandwidth change, because the bandwidth change request (pilot packet) is initiated by the traffic source only. The bandwidth requirement is considered to be a fixed value at any time period. In other words, the bandwidth requirement is not negotiable.

The feasibility of a switch for multilevel control has been studied, for example, by Pattavina [56], in which he also proposed a new policy as outlined below.

Multichannel Allocation

Pattavina's multichannel bandwidth allocation scheme [56] allocates some bandwidth for each call at setup time, and also keeps some reserve channels dynamically allocated to in-progress calls on-demand within the same VP, or channel group (Pattavina's terminology). It is very similar to Hui's multi-level control scheme [31] with the control at setup time in the former being equivalent to call level control in the latter, reserve channel allocation during transmission time to burst level control, and slot allocation of reserve channels during transmission time to packet level control. Only the implementation feasibility at the switching fabric was studied, not the network management and signaling protocol issues.

The scheme considered in [72] is very similar to Pattavina's policy, but is at the VP level instead of the VC level. Each VP is allocated some small amount of bandwidth. The remaining bandwidth

is dynamically allocated and released to VPs on-demand. The analysis shows that the blocking rate and throughput can be improved by as much as 20%, compared to the general allocation for two VPs on a single link.

Admission Control for Compressible Data

The admission control scheme proposed in [41] is based on the assumption that data from all sources can be compressed when necessary. Complete partitioning of bandwidth is assumed. If the original high bandwidth partition is used up, the scheme allows a call with high bandwidth demand to be compressed to a low bandwidth call and then accepted. Poisson arrival process and exponential holding times are assumed for the analytical model. A compressed call is assumed to have the same exponential holding time distribution as the new class. The analysis showed call blocking can be reduced for high bandwidth calls, and increased for low bandwidth calls.

Distributed Source Control

Bursty data with long interarrival times and short burst lengths has been considered as not difficult for traffic control, because any overload resulting from statistical multiplexing can be successfully smoothed out with a moderate amount of buffering at each node. The more serious situation is data streams with long burst lengths.

The essence of distributed source control (DSC) [58] is to break large bursts, generated by the end-to-end window, W_E, over every round-trip delay, T_R, into small burst, using a smaller (smoothing) window, W_S, and shorter periods, T_S, such that the throughput attained on the VC is $\overline{\lambda} = W_E/T_R = W_S/T_S$.

A new call is accepted by DSC if and only if both the throughput (bandwidth) constraint and the buffer size constraint are satisfied. The throughput constraint is given by $\overline{\lambda} + \sum \lambda_i < \lambda_L$, where $\sum \lambda_i$ is the sum of the average bandwidths of in-progress calls, and λ_L is the link capacity. The buffer capacity B should be $B \geq W_S + \sum W_S^i$. Performance is measured by simulation.

DSC assumes that new arrivals may accept reduced bandwidth and that their bandwidth allocation can be increased in duration whenever possible. The setup negotiation is proposed in the following way. A call is offered a reduced window when its original request is denied at setup stage. This call has the choice of accepting this reduced window or looking for an alternate path. A call using a reduced window is called a throttled call. When excess bandwidth becomes available upon call departure, throttled calls will receive additional bandwidth.

The problem with this negotiation scheme is that one extra round of signaling is required to convey the call's acceptance or rejection of the offered bandwidth. In addition, the bandwidth has to

be reserved at all nodes along the path while waiting for the call's response to the negotiation offer. Although it does allow bandwidth increase of some in-progress calls, DSC does not allow bandwidth reduction of in-progress calls, i.e., bandwidth relinquishment from in-progress calls. Similarly, the network can offer a lower bandwidth with DSC, but not a higher bandwidth than the original request of the call.

Traffic Enforcement

Although this thesis does not consider traffic enforcement at the cell level, reactive schemes for traffic enforcement can still offer inspirations to access control policies.

Traffic enforcement is essentially traffic control so that the rate will not exceed a limit. The leaky bucket policy and its variants seem to have gained wide acceptance. The token pool capacity and its renewal period determines the maximal transmission rate or bandwidth a connection can use.

The original leaky bucket policy denies cell admission when the tokens are exhausted, i.e., the agreed upon bandwidth is exceeded. A job queue was suggested to hold the excessive cells [13] [5]. Another improvement was considered by Bala *et al.*[2] to mark the excessive cells for later discarding if and when congestion occurs inside the network. This policy implies a variable bandwidth allocation in the sense that traffic exceeding the initial bandwidth allocation is allowed, but not guaranteed of reliable delivery by the network.

Okada *et al.* [55] first used the dynamic reactive allocation concept in leaky bucket policies. It renews the token pools dynamically according to the network load, with a conventional limit as the lower bound. Compared to virtual leaky bucket, extra gain is obtained by not discarding cells even when the traffic exceeds its limit, as long as such overload does not jeopardize other traffic. Therefore it can improve the network performance over other policies. However, it relies on a ring structure to detect the traffic load, and hence cannot be applied to B-ISDN with general architectures. The idea of reactive allocation is nevertheless refreshing for B-ISDN congestion control.

2.4 Summary and Interesting Problems

There are many challenging access control problems in B-ISDN; three of which are summarized here. These challenges motivated the new access control scheme presented in the remainder of the thesis.

2.4.1 Reactive Control Versus Preventive Control

Many people considered reactive control to be unsuitable for B-ISDN. For example, Hong and Suda [29] cited high data loss at high transmission speed during reaction time as the reason. This thesis considers reactive control to still be viable for B-ISDN for the following reasons:

- The congestion node could discard cells from some selected virtual paths only, as opposed to all paths, so that the amount of retransmission would be reduced. The selected paths could be the offensive paths, the lower priority, or randomly chosen ones.
- The effective region of a reactive control should be of reasonable radius from the congestion node. For instance, it is impractical to expect a node in Vancouver to react to congestion in Halifax.

Take the same network considered by Hong and Suda [29] as an example. The propagation delay from Edmonton to Calgary is 1 ms on a 1 Gb/s regional line. Only affected connections will experience merely 2 ms extra delay due to retransmission, that amounts to 0.2 Mb re-transmission data if 10% of all connections are affected. Suppose those affected connections are from 66 sources (each transmitting at 1.5 Mb/s on average), then a buffer of approximately 3 Kb is required for each source's re-transmission which is quite nominal today where many handheld calculators have 64 Kb storage.

This thesis will attempt to avoid long term congestion through preventive access control, to rely on traffic enforcement, shaping and buffering for short term congestion, and to use reactive access control to resolve medium term congestion.

2.4.2 Call Setup Queue

"Very often, we may tolerate a delay in the resource allocation process without being considered blocked," Hui stated in [31]. Buffering for an entire packet and burst has been considered feasible in access control and bandwidth allocation studies. Setup requests for connection oriented calls, however, are not commonly permitted to be queued; that is, when an arriving call finds no resources available for immediate acceptance, it is blocked. Weinrib and Gopal allowed limited waiting for circuit-switched calls in their routing study [73].

This thesis will consider a finite call-request holding queue available for the purpose of admission control. The rationale is the following. As Weinrib and Gopal stated in [73], the more powerful hardware and faster signaling system significantly reduces call setup time. Shortening the call setup delay below a few seconds becomes unobservable and irrelevant to the human users. The setup time saved can then be used for limited waiting to improve the network performance. This study

also takes the following practical reasons into consideration. It has been demonstrated in daily life that callers sometimes are willing to be put on hold when the called parties are not immediately available. The same philosophy can be used for call admission as well. When a network resource is not immediately available, some callers are willing to wait for service, as opposed to abandoning the call attempt. Furthermore, for non-human users, the call setup delay and transmission delay are not always distinguishable. The communicating users are concerned with the total elapsed time between the instant the call request is issued to the network and the instant the call is finished. The exact breakdown between call setup time and transmission time is irrelevant. Thus, it is worthwhile exploring if more time could be saved by waiting for higher bandwidth. To this end, this thesis considers networks with and without a waiting queue.

2.4.3 Negotiable Bandwidth

The allocation of network resources for traffic using fixed rate coding has been considered as straight forward [58]. One assumption has been that the transmission rate is constant and there is only one rate to choose from.

Access control decision in most policies is either acceptance or denial of the call request. No existing study considers bandwidth negotiation. The call request has been considered to be of a fixed bandwidth. Moreover, the bandwidth allocation is not changed for the call duration. For example, if the leaky bucket policy is used, the token pool size is considered to be constant for the duration of the call [5]. That means the maximal bandwidth usage allowance is remained the same during the call duration.

There have been two studies allowing some degree of flexibility in bandwidth allocation. The first one is the hierarchical channel access policy by Kraimeche and Schwartz in 1987 [41]. The second one is the distributed source control [58] proposed in 1990 by Ramamurthy and Dighe. This thesis considers a situation where the bandwidth allocation is more flexible.

In summary, there are many interesting problems remaining to be studied. The next two chapters of this thesis propose, respectively, a new service and the protocol based on the negotiation and re-negotiation capability of B-ISDN, followed by performance studies.

Chapter 3

Flexible Bandwidth Service and Protocol

A new service class and its associated protocol are proposed at the AAL level and the network layer level in this chapter. The new service class is based on the facts that many digital communications may be carried out with varying bandwidth and that a user's communication demand usually is a range constrained by the minimal performance requirement and the maximal affordability. The objective of the proposed flexible bandwidth service class and protocol is to achieve low blocking rate and high bandwidth utilization/revenue, while keeping the call delay as short as possible.

The first section examines the feasibility of flexible bandwidth communications from the perspectives of both user and equipment capability. The second section provides the motivation for the new service by discussing potential benefits of flexible bandwidth allocation. The third and fourth sections deal with the detailed implementation issues. The service definition document and protocol definition document for the new flexible bandwidth service are provided in these two sections, respectively,

3.1 Feasibility of Flexible Bandwidth Service

The new service is based on the fact that digital communication at the call level may be carried out with varying bandwidth, due to the following two reasons.

1. Communication users can be flexible in their demand, and
2. Communication equipment can support multiple transmission rates.

3.1.1 Traffic Controllability: User's Prospective

As discussed in Chapter 1, the bandwidth demand for many communication applications is in fact a range limited by the minimal performance requirement at one end and the affordability or end equipment capacity at the other. Hence, any bandwidth within this range should be acceptable to the user.

Call requests occasionally have to be blocked to prevent network congestion and maintain the quality of service of established connections within acceptable ranges. Given the options, many users would choose lowering their performance demands rather than being denied service altogether. When a video-phone call request gets a busy signal, for example, the caller might be willing to make a voice phone call, rather than waiting to try the video-phone again. If a realtime remote file fetch cannot be performed due to the lack of bandwidth, a computer programmer may initiate a low bandwidth file transmission as a background process on his workstation and start attending some other programming tasks, instead of retrying over and over again. Flexibility of users' demands, however, must be supported by the communication equipment.

3.1.2 Traffic Controllability: Equipment Capability

Bala, Cidon and Sohraby acknowledge that data traffic can usually be slowed down in order to cope with network congestion [2]. However, when it comes to congestion control, most attention has been given to uncontrollable realtime traffic. Voice and video are often used as examples to show that the majority of traffic types cannot be slowed down [24].

However, new hardware technologies are emerging to make it possible to change the customer premise equipment (CPE) communication bit-rate during the transmission. One example could be the single-chip half-duplex multi-rate transcoder described in [46]. It is pin selectable to transmit at 32 kb/s with adaptive differential pulse code modulation or at 16 kb/s with subband coding. A voice call can, hence, change its transmission rate (bandwidth) on the fly with this device.

Another example could be the multi-rate random access memory device described in [68]. It features a pin selectable 32 or 8 bits wide input/output (I/O) interface with I/O bandwidth up to 6.46 Gb/s. Using this device, a broadband data communication can be carried out with different bandwidth.

Moreover, up to half of the B-ISDN traffic will be inter-LAN communication [67] with the remaining traffic also including many other data communications. It has been recently recognized that the primary driver for B-ISDN development is LAN interconnection [70], although video and voice are widely considered when it comes to access control and bandwidth allocation studies. LAN gateways or routers usually provide a range of transmission rates to choose from. Therefore, the

majority of B-ISDN traffic is flow controllable from the equipment's perspective.

3.2 Possible Benefits of Flexible Bandwidth Service

The important performance measurements are the bandwidth utilization (or revenue) and the call blocking rate.

Each call generates revenue at a rate that is dependent on its class and the actual usage of bandwidth. The network may keep revenue (or utilization) as high as possible by assigning to each call the highest bandwidth the customer is willing to pay. The consequence, however, is the potential increase in blocking rate. The more the bandwidth assigned to ongoing calls, the less the capacity left to accommodate future call arrivals. On the other hand, the more the bandwidth reserved for future arrivals to ensure low blocking rate, the lower the network utilization and revenue are.

Essentially the proposed new service solves this dilemma in the following ways. Whenever possible, all calls get the highest bandwidth the users are willing to pay for. However, when an arriving call cannot be accommodated otherwise, the network will notify some in-progress calls to voluntarily relinquish some bandwidth in order to accommodate the new call. In essence, the bandwidth relinquishment by in-progress calls is voluntarily, because these calls have registered with the network their willingness to give up bandwidth up to a limit at call setup stage. At call setup stage, flexible calls, on the other hand, also have notified the network that they are ready to increase their bandwidth usage up to a limit. All the in-progress calls are serviced still within the two limits that they have specified as part of the call setup request message.

The following two scenarios illustrate how the proposed new service can improve network performance.

Scenario 1: Call Level

In this section, the example broadband network is a virtual private B-ISDN for the Alberta medical community, as shown in Figure 3.6. This imaginary Alberta Medical Network (AM-Net) connects all hospitals, clinics and government agencies in four Alberta cities through a virtual private B-ISDN on a public telecommunication network operated by the Alberta Government Telephone (AGT). The AM-Net also provides broadband communication services to those users on virtual metropolitan area networks (MANs) in Edmonton and Calgary.

Suppose that the virtual path between the University of Alberta Hospital and the Grey Nuns Hospital has a maximal capacity of VT1.5 (1.728 Mb/s), where a voice call requires 32 kbit/s bandwidth.

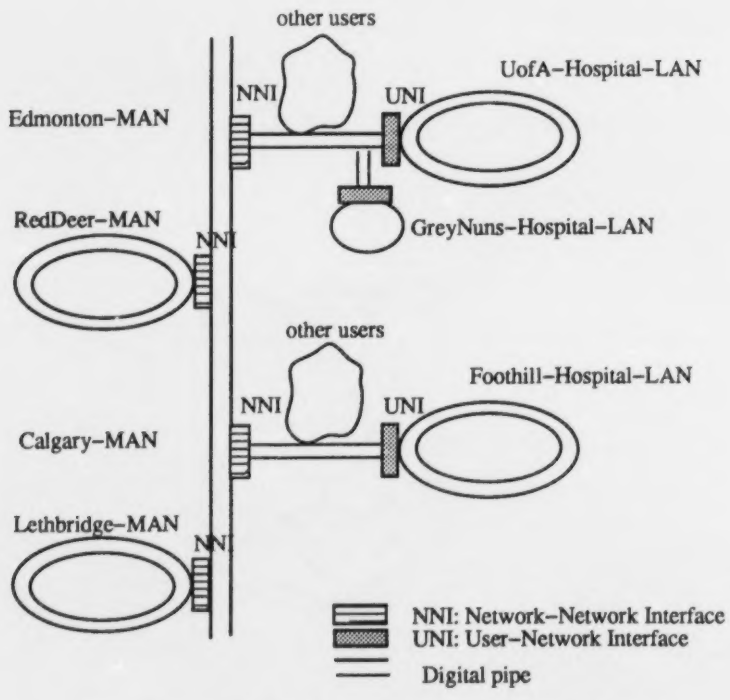

Figure 3.6: Example Network: Alberta Medical Network

Consider the case where a physician at the Grey Nuns Hospital needs to phone a radiologist at the University of Alberta Hospital, and only 16 kbit/s bandwidth is available on the VP. With conventional service classes, this call cannot be accepted without increasing the VP's capacity, since bandwidth is insufficient to ensure the required communication quality on the VP.

This call request can be accepted, however, without changing the VP's capacity, if it or some in-progress calls support flexible bandwidth allocation by, say, using the selectable-rate codec described in [46]. Two possible solutions then exist, depending on which call is a flexible call.

If the physician has a phone with selectable rates, the UNI will simply accept this call and allocate 16 kbit/s to it, knowing that the arriving flexible voice call requires bandwidth between 16 kbit/s and 32 kbit/s.

If the physician's phone does not have selectable rates, or only high quality voice service is required for the conversation between the physician and the radiologist, the UNI will request an ongoing flexible call to switch its transcoder transmission rate to 16 kbit/s. Then, the physician's call can be accommodated at 32 kbit/s.

The first solution is essentially bandwidth negotiation at the call setup stage. It can be ac-

complished by out-band signaling. The second solution involves bandwidth relinquishment for in-progress calls. In either case, the new service can achieve potentially better resource utilization and call blocking rates than those achieved by conventional services.

Furthermore, whenever new bandwidth is released upon completion of an existing call, additional bandwidth can be allocated to the flexible call which is currently using 16 kbit/s bandwidth so that, high quality transmission can be regained without compromising the network performance.

The scenario above considers flexible bandwidth allocation at the call level without changing the virtual path capacity. The next scenario applies the same concept to the network level in managing the virtual path.

Scenario 2: Network Level

This scenario considers the access control and bandwidth allocation problem at the network level on the link between Edmonton and Red Deer in our Alberta Medical Network example.

The underlying link capacity is assumed to be an STS-3c. Suppose there are two virtual paths existing on that link: one from Edmonton to Red Deer at STS-1, and the other from Edmonton bypassing Red Deer to Calgary at STS-2. Moreover, the first one carries mainly video transmissions for medical education. The second one contains a consultation session on a patient's chest radiology examination between a radiologist in Edmonton and a physician in Calgary. The network has allocated half of an STS-1 link capacity to this call, based on the normal traffic pattern for such a call.

Now, suppose the case turns out to be more complicated than the physician initially thought, and two more sets of images have to be added to the display by the physician. The virtual path can handle only one set within the required 2 seconds delay limit, without degrading the service. Under existing services, the performance requirement by the doctors cannot be met. It could, however, be met with the flexible bandwidth allocation service.

Now, let us assume the educational video communication has been specified as a flexible call, with the acceptable bandwidth between one STS-1 for full motion video and half STS-1 for slow scan video.

With the flexible bandwidth allocation capability, the radiology consultation call issues a new request to the virtual path for doubling the bandwidth allocation, when the physician clicks the button to choose two images. The virtual path, being already fully allocated, turns to the network for a bandwidth increase by half an STS-1 bandwidth. The network then relinquishes a half STS-1 from the educational call and assigns it to the requesting virtual path. Two seconds later, the image transmission is completed. After the radiology consultation call detects low bandwidth usage for a

sufficient period of time, it requests bandwidth allocation reduction to the virtual path, which in turn indicates this to the network. The network then re-allocates bandwidth between the two virtual paths and the education video returns back to full motion after a couple of seconds of slow-scan transmission.

3.3 Flexible Bandwidth Service Definition

In defining the flexible bandwidth service, the following two assumptions are made:

- An acceptable bandwidth range will be specified by the call either as part of the setup request message, at any time in the call duration, or as part of the service profile entered at the UNI at subscription time;
- The network has adequate signaling ability to convey the negotiation and allocation information.

The protocol service definition documents specify the service provided by a layer to its user in terms of service primitives with the associated service parameters, and the interrelationships between the primitives [28]. This section provides the service definition document for the flexible bandwidth service. The first part defines the service primitives. The new parameters are described in the second part of this section, which is followed by the time sequencing of the primitives.

3.3.1 Service Primitives

In addition to the existing six connection-oriented network layer services and four transport layer services defined in ISO 8348, two new services are introduced in this thesis for the flexible bandwidth service class at these two layers. They are

- Negotiation service
- Allocation service

Each service has four primitives: request, indication, response and confirmation. They are expressed, according to the ISO naming convention, as

- N-Negotiate.request,
- N-Negotiate.indication,
- N-Negotiate.response,
- N-Negotiate.confirmation,
- N-Allocate.request,

- N-Allocate.indication,
- N-Allocate.response, and
- N-Allocate.confirmation.

where the prefix "N-" indicates that the primitives belong to the network layer.

The negotiation request primitive is a confirmed primitive; in other words, it must be acknowledged. It is used to negotiate the parameters of an existing call, instead of establishing a new call.

The Allocate primitive is initiated by network switches to change the bandwidth allocation of an existing call. The allocation request does not have to be acknowledged. A positive bandwidth allocation parameter value indicates the increments of bandwidth allocation, while a negative value indicates decrements. In the remaining part of this thesis, the term "re-allocation" will be used specifically for bandwidth allocation increment for in-progress calls, and "relinquishment" specifically for bandwidth allocation decrement. As far as the protocol is concerned, however, they use the same allocation service at the network layer, the only difference is the sign of the allocation parameter value.

3.3.2 Service Primitive Parameters

Table 3.2: New Service Parameters for Flexible Call Class

Service Primitives	Min. and Max. Bandwidth	Allocated
Connect.request	expected bandwidth range	preferred
Connect.indication	expected bandwidth range	allocated
Connect.response	expected bandwidth range	local decision
Connect.confirmation	-	final allocation
Negotiate.request	expected new range	-
Negotiate.indication	expected new range	allocated
Negotiate.response	expected new range	local decision
Negotiate.confirmation	-	final allocation
Allocate.request	-	bandwidth change
Allocate.indication	-	bandwidth change
Allocate.response	-	bandwidth changed
Allocate.confirmation	-	bandwidth changed

At both the network layer and the transport layer of the OSI model, a new set of quality of service parameters are added to the connection service primitives for the flexible call class. They are the minimal, maximal and allocated (preferred) bandwidth. All the new primitives for negotiation and allocation use the same parameters as the connection service. The meanings of these new quality

of service parameters are tabulated in Table 3.2 for each primitive, where the entry "-" indicates this parameter does not have specific meaning for that primitive.

The use of "allocated" parameters in Connect and Negotiate services at the network layer can be illustrated by the following process of call establishment. The primitive Connect.request specifies the bandwidth range expected for the session/call. It is propagated from the call origin to the destination. A network node along the route will receive the decision made by previous (upstream) nodes as well as the original expected bandwidth range. It will make its own decision based on local resources and previous allocations for that call. If its decision is to accept the call, then it will replace the new allocation value and pass the primitive to the next node along the route.

The "allocated" parameter is used in Allocation service to indicate the amount of bandwidth changes for the addressed call, and acknowledgement the specified bandwidth change has been completed.

The bandwidth range or flexibility are specified by CPE as a maximal bandwidth and a minimal bandwidth, as the lowest bandwidth it can accept and the highest bandwidth it can afford. The call is expected to be accommodated with bandwidth between the specified maximal and minimal bandwidth values. Let b_{max} and b_{min} denote the maximal and minimal bandwidth values specified by a call. If the maximal bandwidth equals the minimal bandwidth ($b_{min} = b_{max}$), then that call is a conventional one, i.e., with fixed bandwidth requirement; otherwise it is referred to as a ***flexible call.***

The relation of these primitives is defined in the next part.

3.3.3 Time Sequence of the Primitives

The interrelationships of these primitives are illustrated by the time sequence diagram in Figure 3.7. This time sequence diagram depicts a successful request only. The unsuccessful situation is simply handled with disconnect primitives, instead of the normal primitives. If the connect request is unsuccessful, for instance, a disconnect primitive will be sent back from the network node where the blocking decision is made.

The re-negotiation can be initiated by either the calling party or the called party using Negotiate primitives. This process is very similar to the process of call setup, except the outcome is not call establishment, but a new set of bandwidth parameters.

The allocation request can be initiated by the network as shown in Figure 3.7. It can be used by the network to relinquish bandwidth from in-progress calls within a pre-agreed upon bandwidth range; that is, the bandwidth after relinquishment is no less than the minimal bandwidth requirement specified in the previous Connect or Negotiate primitives. The allocation can also be used by the

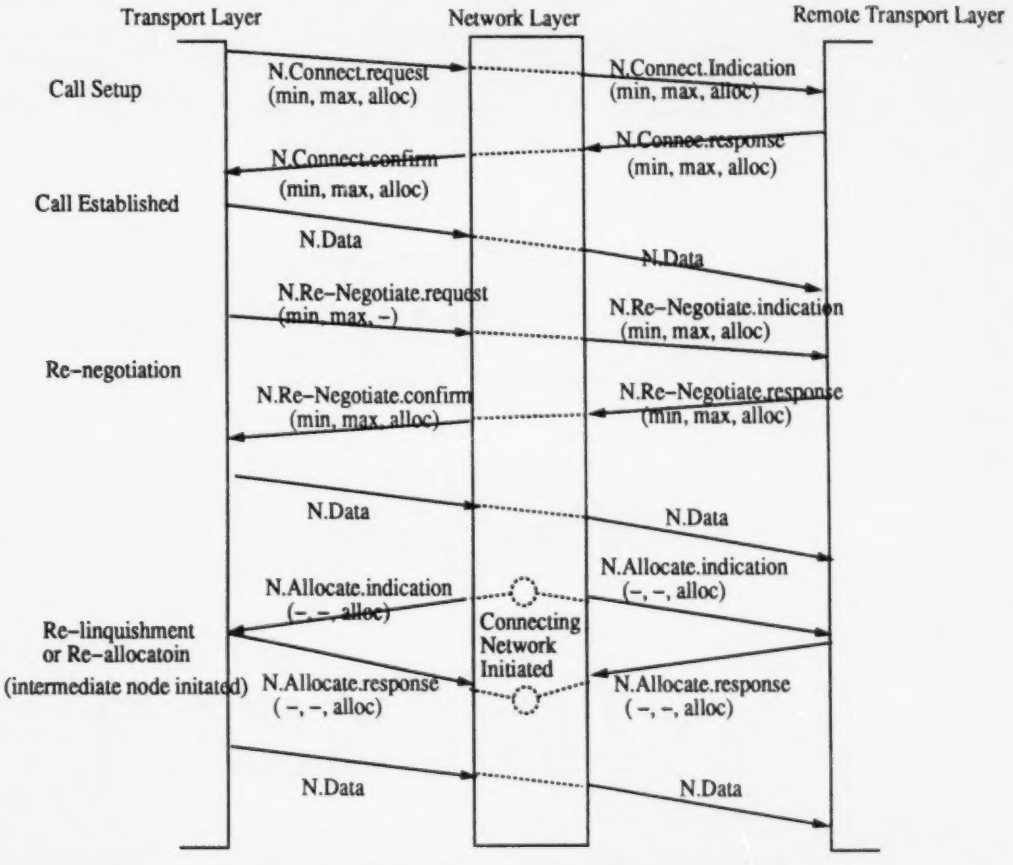

Figure 3.7: Time Sequence Diagram for Flexible Bandwidth Service at the Network Layer

network to increase the bandwidth allocation of in-progress calls up to their specified maximal bandwidth. The end user can optionally acknowledge it. The originator of an unacknowledged allocation request can assume its completion after a certain timer duration which is network specific and may vary from node to node. A reasonable value would be the maximal round-trip delay on the radius of the effective region of reactive control.

3.3.4 Services Required

No new services from the data link layer are required to support the new services and primitives at the network layer. The new network layer services required by the transport layer have been defined in subsections 3.3.1 and 3.3.2.

3.4 Protocol Specification

3.4.1 Protocol Data Units

No new type of protocol data unit (PDU) is required by the proposed new service. All necessary PDUs are based on those defined in CCITT Recommendation Q.931 [7] with new configurations and meanings defined below.

PDUs for Call Establishment and Negotiation

A SETUP message is similar to that defined in Q.931. But, as opposed to only one in Q.931, up to three Bearer Capability (BC) information elements (IEs) are allowed to identify the previously allocated bandwidth, and the minimal and maximal requirements. If service profile is implemented, then BC IEs can be absent, and the default bearer capability request is stored in the service profile record.

A BC IE should be present in the CALL-PROCEEDING message responding to a SETUP message containing three BC IEs. This BC IE indicates the final allocation by the network and the called party. The call will proceed with the bandwidth indicated by this BC IE.

If the network or the called party does not support flexible calls, then the responding CALL PROCEEDING message should include a Cause IE with cause value 65 (bearer capability not implemented), when receiving a flexible call setup request.

PDUs for Re-Negotiation

STATUS ENQUIRY and STATUS messages are used for bandwidth re-negotiation for in-progress calls. To do so, three BC IEs are added to STATUS ENQUIRY for the minimal, maximal requirement and the allocated capacity, and one BC IE is added to STATUS to indicate the final allocation. The STATUS ENQUIRY message can be sent through the selective broadcast signaling virtual channels to those calls whose service profiles match the re-allocation criteria, so that only one signaling message is necessary to convey the information to all intended recipients.

PDUs for the Allocate Service

A NOTIFY message with BC IE is defined by CCITT as the network's indication to the user for a change of the bearer capacity in ISDN. It can be used for the same purpose in B-ISDN; that is, to transport the Allocate service for the flexible calls. It can be sent through the selective broadcast

signaling virtual channels to those calls whose service profiles match the re-allocation criteria, so that all calls can be notified with only one broadcast message.

A special case of allocation is call suspension and resumption, i.e., the complete relinquishment and re-allocation of a call's bandwidth. In this case, the existing SUSPEND and RESUME messages in Q.931 and HOLD and RETRIEVE in Q.932 can be used directly.

3.4.2 Signaling Protocol Details

A network layer (layer 3) signaling protocol is designed to implement the flexible bandwidth service at the UNI. Layers 2 and 1 protocols will be the common ATM and SONET protocols, respectively. It is the layer 3 protocol that differentiates a flexible call from a conventional non-flexible call.

This layer 3 protocol is described in CCITT Specification and Description Language (SDL) in Figure 3.8 and Figure 3.9 for the user side and network side of the UNI, respectively. The protocol for the network-network interface (NNI) can be derived from them. This layer 3 protocol contains the essential signaling for the flexible service only. It does not have, for example, the timeout recovery and handling of invalid incoming messages. The omitted parts are common to communication protocols, and do not affect the feasibility of implementing the proposed new service and its admission control and bandwidth allocation scheme. The remaining part of this section navigates through the protocol.

When receiving a connect request service primitive from higher layers, the layer 3 user protocol sends out a SETUP message, and waits for a CALL PROCEEDING message from the network. The SETUP contains BC IEs indicating the minimal and maximal transmission rates. The minimal and maximal transmission bandwidth is passed down to the network layer from the transport layer where this range is determined according to performance requirements such as maximal transmission time, maximal affordable bandwidth, and estimated amount of information.

If a DISCONNECT message, instead of the CALL PROCEEDING, is received, the call attempt fails and the call is blocked. Upon receiving a CALL PROCEEDING message, the connection is established, and data can now be transmitted at the rate indicated in the BC IE in the received CALL PROCEEDING message.

During the call, upon receiving a NOTIFY message, the user shall start using a new transmission rate indicated in this message by the network. On the other hand, the user can also request a change in current transmission rate by sending a STATUS ENQUIRY message. The network will respond with a STATUS message for its approval or denial of the negotiation. When a negotiation is denied, the user has the option either to proceed with the current service contract or to abandon the call.

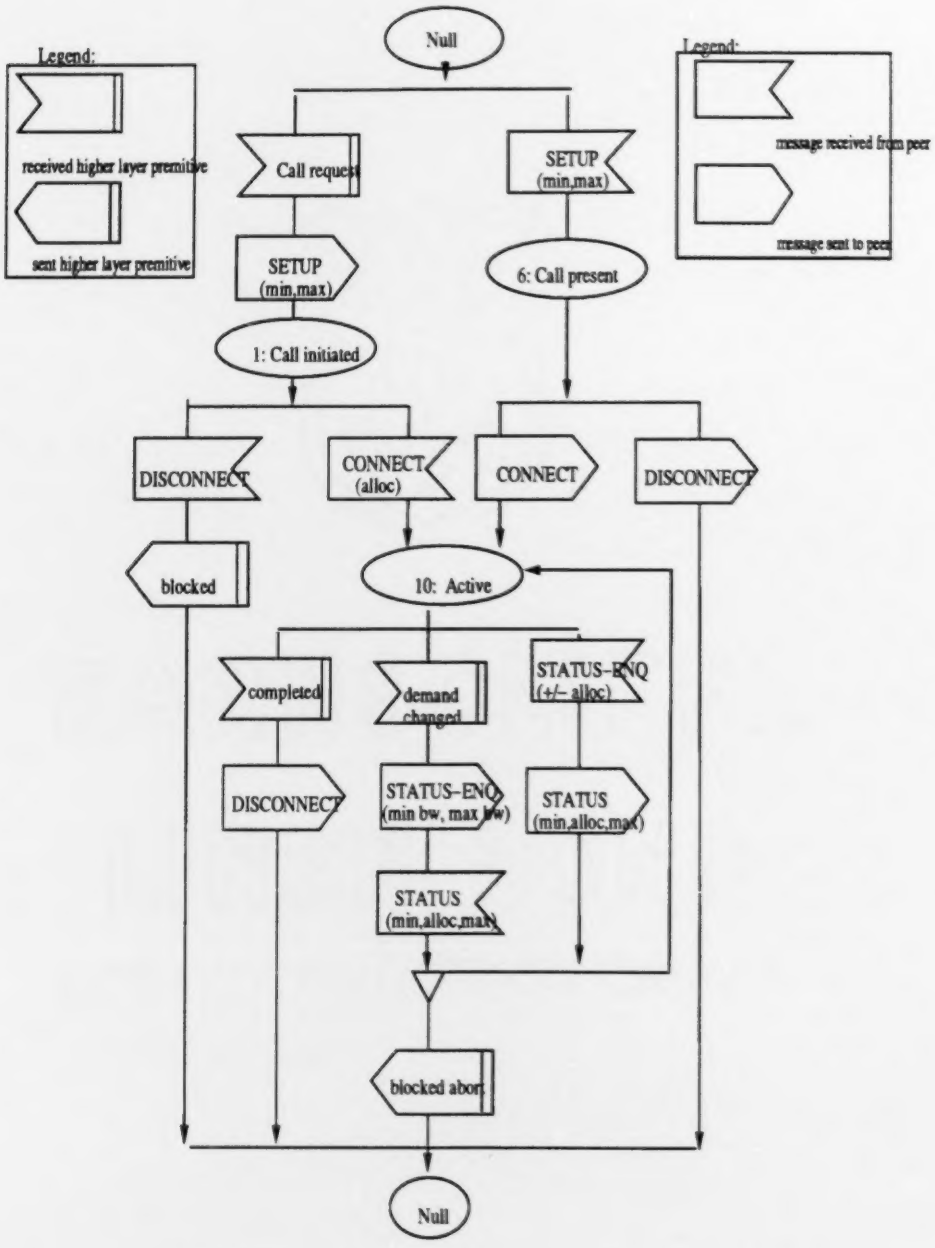

Figure 3.8: Flexible Call Control Protocol at User Side of UNI

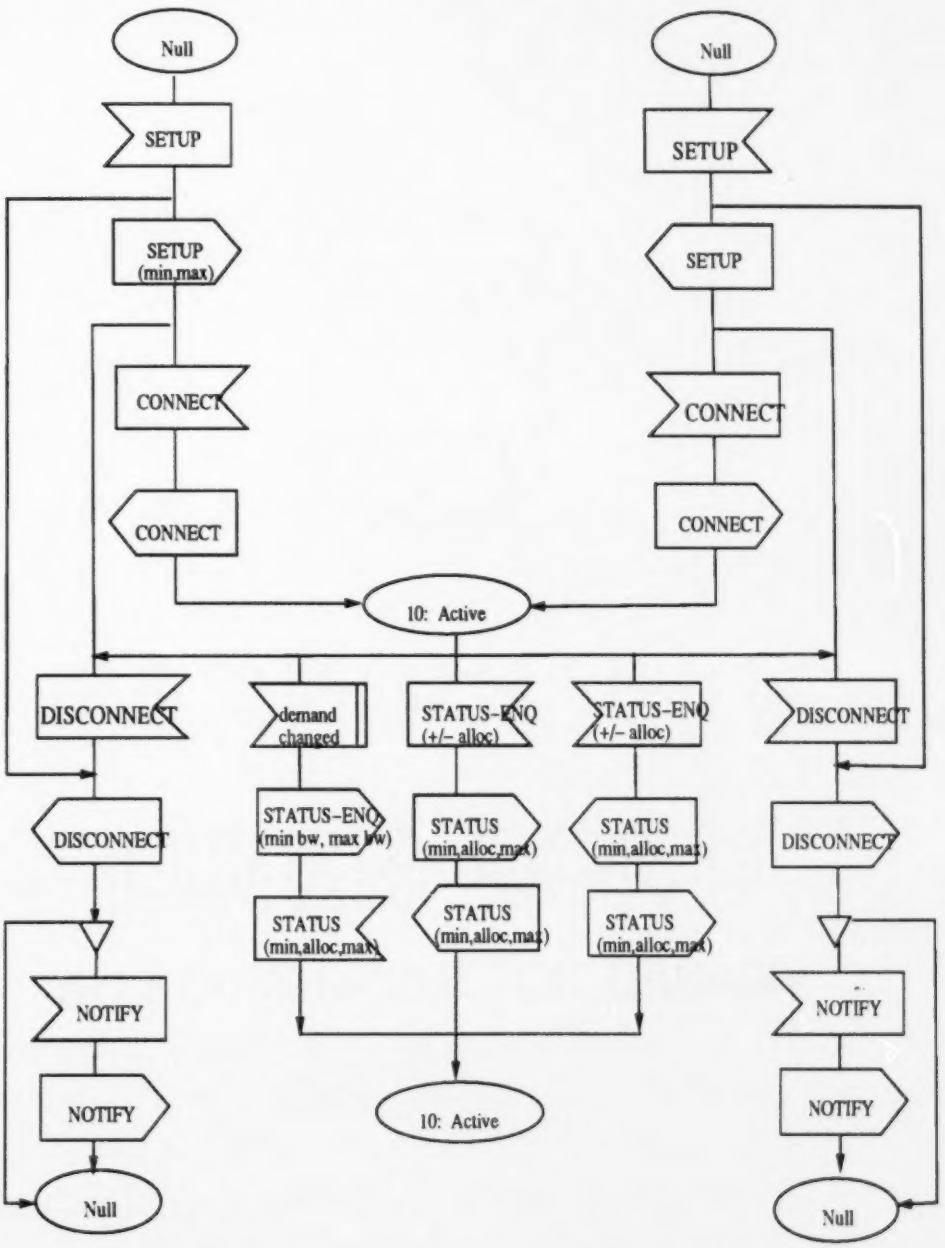

Figure 3.9: Flexible Call Control Protocol at Network Side of UNI

Upon completion of the call, a DISCONNECT is sent by the user to notify the network to release the connection.

The network layer protocol at the called party will respond to an incoming SETUP with a CALL PROCEEDING containing its choice of transmission rate in the BC IE, and get ready for receiving data. During the call, connection can be requested to change its transmission rate, or it may initiate a change request itself, similar to that part of the protocol described above for the calling party.

The network side of the layer 3 protocol is very similar to that of the user side, expect that it also passes the various request messages between UNI and NNI, with its own decision added onto these messages and a more stringent bandwidth range if and when necessary.

The network access controller at UNI will assess the resources to make a decision on whether or not to admit a call. A call can either be accepted, or blocked due to a lack of resources. For the accepted call, the network access controller will then determine how much bandwidth will be allocated to it, and may need to relinquish bandwidth from some in-progress calls which are currently allocated more bandwidth than their specified minimal value.

Upon the completion of a call, the network may decide whether or not to re-allocate the newly released bandwidth to the in-progress calls. The bandwidth allocation can be negotiated at the call setup stage, or changed dynamically in progress on demand, or both. Hence, some calls can have the bandwidth allocated to them reduced to some extent if the network is overloaded. On the other hand, calls may use more bandwidth than normally allowed if the network is relatively idle. This bandwidth negotiation and allocation scheme is discussed in the next chapter.

3.5 Compatibility with Existing Protocols

The B-ISDN signaling protocols are considered to be derived from existing ISDN protocols. An important issue then is protocol compatibility; that is, the newly derived protocol should be able to coexist with the existing protocols without affecting the network operation. Because any practical network evolution is a gradual process, existing networks cannot be replaced overnight, new and old ones must be able to interwork.

The flexible bandwidth service and its protocols designed in this chapter are compatible with both existing network facilities and existing user premise equipment.

If the network does not have the necessary signaling or processing facility, then the bandwidth range specifications for a flexible service will not be processed; that is, the first and third BC IEs are ignored. The access control and bandwidth allocation will be conducted as it is today. Hence, the existing signaling network is compatible with the flexible bandwidth service, as it defaults to flexible

bandwidth service with the maximal and minimal bandwidth demands being the same value.

If a CPE does not have the flexible call capability, a network with the new signaling protocol is still able to accommodate such equipment. In the absence of bandwidth specification, a call will be treated as a conventional one without flexibility; that is, as if all the three BC IEs are identical. The signaling network will process PDUs as if it does support the new service. Hence, the existing equipment is compatible with the new signaling protocol.

Chapter 4

FACT: the Flexible Access Control Technique

A new admission control and bandwidth allocation technique is proposed in this chapter. It is referred to as Flexible Access Control Technique (FACT). Besides a basic policy, five variants of the basic policy are also proposed to address when and how bandwidth should be relinquished from in-progress calls.

4.1 The Basic Policy

The basic policy with FACT can be stated as follows. Let C be the available bandwidth, $(b_{k,min}, b_{k,max})$ the bandwidth range requested by a call k, b_i the bandwidth allocated to an in-progress call i, I the set of in-progress calls, and j the arriving call numbered.

The call admission rules for FACT are the following:

Rule 1: Accept the arriving call without changing in-progress calls, if $b_{j,min} + \sum_{i \in I} b_i \leq C$

Rule 2: Re-allocate in-progress calls, and then accept the arriving call, if $C - \sum_{i \in I} b_i \leq b_{j,min} \leq C - \sum_{i \in I} b_{i,min}$.

Rule 3: Block or hold the arriving call in a waiting queue, if $b_{j,min} + \sum_{i \in I} b_{i,min} > C$

Rule 1 considers the situation where there is enough free bandwidth available to accommodate the arriving call. In this case, the arriving call j will be given bandwidth b_j, $b_{j,max} \geq b_j \geq b_{j,min}$, the exact value of which is to be studied in this thesis.

Rule 2 deals with the situation where there is not enough bandwidth available for the arriving call, but the arriving call can be accepted after relinquishing bandwidth from in-progress calls

according to their prior consent. Several difference relinquishment policies are possible. They are discussed in the next section.

Rule 3 states that a call request has to be rejected or postponed if the minimum bandwidth demand of all in-progress calls together with that of the arriving calls exceeds the capacity, which means the call cannot be accepted even after the relinquishing maximal amount of bandwidth from the in-progress calls.

With FACT, in-progress calls are notified of the re-allocation through outband signaling, should the re-allocation be necessary, and consequently change their transmission rates to the new values which are still within the range which they have agreed upon. A choice of the bandwidth allocation b_i will be sent back to the call i source by the network, probably through the selective broadcast signaling virtual channels [8]. The users then proceed with their calls at this negotiated bandwidth b_i's.

The transmission requirements of a call may also be changed during the progress of the call for reasons such as approaching real time deadlines, change of delay requirements, human user taking a coffee break, detection of an emergency and so on. FACT can also support such realtime dynamic bandwidth re-negotiation. For instance, realizing a deadline, a call sends out a new (higher) bandwidth range specification through the out-band signaling channel. The network will then go through basically the same procedure as dealing with new arriving calls mentioned above. A new bandwidth for the call is, thus, agreed upon by both the network and the call. Upon being allocated the new bandwidth, the call then proceeds at the new higher bandwidth so that the rest of the transmission can be completed before the deadline.

4.1.1 Application of FACT

FACT can be applied at both customer premise equipment (CPE) and network switches.

The network switches do the following:

- Call acceptance/denial based on the service requirement specifications provided by CPE and the ability of the network to meet these requirements along with those of existing calls. If a call is accepted, the service requirement specification becomes a service contract between the end-terminal and the network.
- In-progress call controls involving bandwidth re-allocation within the specifications for in-progress calls, and re-negotiation of service specifications with in-progress calls
- Definition of service calls by the network based on traffic characteristics and performance requirement.

- Means for some selective load shedding by the network to achieve network resiliency, without unduly impacting the service contracts (specifications).
- Means for conveying congestion status and information across the various interfaces.

The controls at CPE involve the following:

- Negotiation of service specifications with the network at the call setup time and within the call duration. The service specifications include minimal and maximal bandwidth requirement, (or equivalently, maximal and minimal delay in addition to estimated amount of information).
- Dynamic control of input parameters based on the service contract from the network (possibly by means of transmission rate selection, traffic-shaping, adaptive window or credit management).

4.1.2 Effective Control Region

When bandwidth negotiation and re-negotiation are used in B-ISDN for congestion control, the affected network nodes and end users must be confined within a region of reasonable distance.

A flexible call's bandwidth allocation is flexible only within a certain region, called *effective control region* for that call under FACT. A flexible call is open to negotiation with every node inside the effective control region. A flexible call outside its effective control region is not considered as flexible. In other words, network nodes outside the effective control region of a call do not attempt to (re-)negotiate bandwidth with that call. Figure 4.10 depicts the effective region concept, where flexible calls have an effective control region of radius 100km. A flexible call originating from Switch C is within its effective control region at Switch D, and, thus, is subject to (re-)negotiation, relinquishment and re-allocation at Switch D. It is however outside the effective control region as far as Switch E is concerned. Thus, Switch E will not relinquish bandwidth from that call.

The radius of an effective control region can be chosen according to link speed, traffic characteristics, response time requirements and other factors such as the buffering capacity at switches. For instance, a high speed network probably should choose a small effective control region, while a network with ample buffer space can afford to have a largeer region.

FACT only relinquishes bandwidth from calls originated and terminated inside their effective control region. That is so because it is unreasonable and impractical to exercise control in response to a congestion taking place across the continent, as discussed in Chapter 2.

Although re-allocation under FACT does not have to be limited by distance, it should still be applied to calls along the same route as the completed call. This is to avoid excessive signaling messages. Otherwise, a re-negotiation signaling message has to be sent to *each* VCC for every call

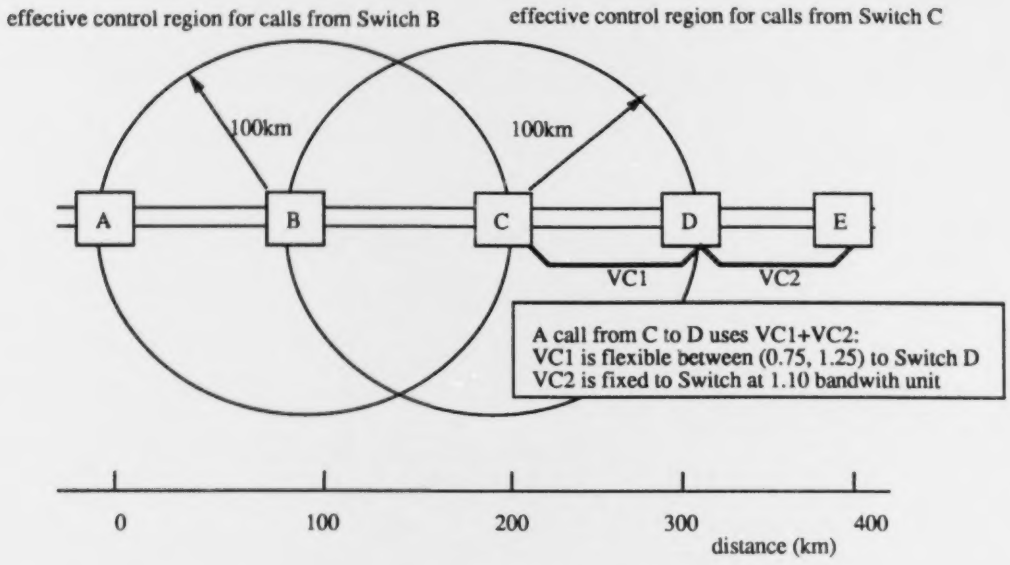

Figure 4.10: Effective Control Region for Flexible Calls

affected. As there is no guarantee that the re-negotiation request can be accepted by other nodes, an acknowledgement message therefore has to be used.

4.1.3 Classification of FACT

FACT is a resource demand reduction scheme. The user permits service degradation and denial in the event of possible congestion, and the former is considered more preferable than the latter.

FACT can be regarded as employing both preventive and reactive control schemes. FACT prevents long term congestion by several means. Firstly, it may deny a network access request to a call. Secondly, it may allocate less bandwidth than the maximal request to an accepted call. Finally, it may also relinquish bandwidth from in-progress calls to accept a call without causing long term congestion. As a result congestion is expected to be very rare. Bandwidth can be relinquished as a reaction to persistent congestion, say, caused by a link failure, or many sources simultaneously transmitting over a long period of time. Moreover, the reactive nature is also reflected by the re-allocation of excessive bandwidth.

FACT is a layered control technique. It is to be used at the virtual path, the virtual channel or at the burst layer. It does not apply to the cell layer where some form of rate control/policing is assumed to exist.

The rest of this section compares FACT to the objectives for congestion control scheme introduced in Section 2.2.3.

- Efficiency: FACT decreases bandwidth allocation when the number of calls is high, and increases it when the number of calls is low, in an attempt to keep the network from being overloaded or underloaded. It utilizes the out-band signaling facilities in integrated networks to adjust bandwidth. The amount of overhead incurred by FACT is the subject of the analytical studies in Chapter 6 and the simulation study in Chapter 9. The results show that, through FACT signaling activities can be frequent, yet the total signaling traffic load incurred is very small, compared to the signaling capacity of B-ISDN.
- Fairness: FACT itself does not impose any restriction on fairness. As discussed in the next section, FACT has both fair and unfair variants available.
- Responsiveness: The key motivation for FACT is to be responsive to network load. It utilizes the out-band signaling in B-ISDN to respond to the network load changes in one definitive step, as opposed to, say, several steps as in adaptive window schemes.
- Robustness: FACT should be able to work well even for unknown traffic situations as will be shown in Chapter 5. Furthermore, it can also handle traffic surges in emergency situation as shown in Chapter 9.
- Global Optimization: FACT is based on the best interest of all subscribers and the network as a whole. Calls co-operate with each other and with the network in their flexibility to relinquish bandwidth for new arrivals and to accept more bandwidth (and finish earlier) if necessary.
- Effectiveness: FACT reduces the possibility of congestion and call blocking by reducing the bandwidth allocation under high loads and shortening the call holding time under light loads.

4.2 Variants of the Basic Policy

As indicated in Rule 2 of the basic policy, there are several possible relinquishment methods, resulting in several policy variants for FACT. This section discusses these policy variants. These variants can be classified into two groups. The first group considers which calls are to relinquish bandwidth and how much from each call. The second group considers when the relinquishment can occur. The total amount of bandwidth relinquished cannot be less than the minimal bandwidth requirement of the arriving call.

4.2.1 Even and Uneven Relinquishment Policies

The first policy is to relinquish equally from all in-progress calls of the same class such that in-progress calls from the same class use the same amount of bandwidth all the time. This policy is referred to as the *even relinquishment* policy. Let N_p denote the number of in-progress calls when the new call arrives and triggers the relinquishment, C the bandwidth. Then, the bandwidth allocation for each call is $C/(N_p + 1)$ after even relinquishment, and each in-progress call has given up $C/N_p(N_p + 1)$. The occurrence of relinquishment implies that $C/(N_p + 1) \geq b_{min}$ and free bandwidth prior to the relinquishment is less than b_{max}.

The even relinquishment policy can ensure fairness among all the calls of the same class, but it does so at the cost of affecting *all* in-progress calls.

The second policy is intended to limit the number of calls affected by relinquishment to a minimum, while compromising fairness. This policy is named the *uneven relinquishment* policy. It relinquishes the necessary bandwidth from a minimal number of calls; that is, each call relinquishes a maximal necessary amount. As a result, the arriving call is always accepted at the minimal bandwidth b_{min}, and the bandwidth usage of those relinquishing calls is also reduced to as low as necessary to accommodate the new call.

If $C - \sum_{i \in I} b_i > b_{j,min}$, then no relinquishment is needed, the arriving call j will be accepted at $\min\{b_{j,max}, C - \sum_{i \in I} b_i\}$. The uneven relinquishment is required when $C - \sum_{i \in I} b_i < b_{j,min}$. In this case, only $b = b_{j,min} - (C - \sum_{i \in I} b_i)$ bandwidth units have to be relinquished. To make the number of relinquishments minimal, this policy requires that the calls to relinquish are those which has the exact, or the closest to the exact, amount of bandwidth available for relinquishment as required. Such a call can be found by keeping in-progress calls (in set I) sorted in ascending order of the bandwidth available for relinquishment, $b_i - b_{i,min}$. The index of the call which will relinquish bandwidth for the new arrival is determined by the largest i satisfying $b_i - b_{i,min} \geq b$. If b cannot be relinquished from one call, then the last k calls in I have to be chosen, and $k = \max\{j : \sum_{i=j}^{N_p}(b_i - b_{i,min}) \geq b\}$, where N_p is again the total number of in-progress calls.

Each of these two relinquishment policy variants has its own merits. A choice has to be made according to performance requirement for a specific network. For instance, if the objective is to keep the number of relinquishments as low as possible, then the uneven relinquishment policy should be used. On the other hand, the even relinquishment policy should be used, if the concern is to treat all calls equally and fairly.

4.2.2 Immediate, Delayed and Early Relinquishment

This group of FACT policy variants deals with *when* to relinquish and re-allocate bandwidth from and to flexible calls. The following three policies are possible:

1. Immediate relinquishment: to relinquish whenever necessary.
2. Delayed relinquishment: to postpone relinquishment by holding calls in a waiting queue until the queue is full, at which stage the relinquishments occur and waiting calls are admitted as a batch.
3. Early relinquishment: to relinquish as soon as the number of in-progress calls reaches one of a few preset limits.

Figure 4.11 depicts the differences among these three policies in terms of call admission and bandwidth allocation.

The immediate relinquishment can probably best utilize the benefits of flexible bandwidth service. The granularity of bandwidth allocation is $(b_{max} - b_{min})/(H - L)$, where $H = \lfloor C/b_{min} \rfloor$ is the maximal number of flexible calls that can be served by the capacity C, and $L = \lfloor C/b_{max} \rfloor$ is the maximal number of flexible calls that can be served before starting relinquishment.

However, every change of traffic can trigger a relinquishment. It may cause an undesirably high load to the signaling channel. The other two policies are designed to reduce the frequency of relinquishments and the amount of signaling.

The delayed relinquishment policy conducts relinquishment only when necessary; that is, when a call otherwise would be lost. In this case, a finite queue of length S is available to hold the calls. The relinquishment is performed only if a new arriving call finds that the queue is full. The granularity of bandwidth allocation is $(b_{max} - b_{min})/s$, where $s = \lceil (H + S - L)/(S + 1) \rceil$ is the number of relinquishments required for the network to reach from fully idle state to service the maximal number of calls possible (including both in-progress and queued calls).

The delayed relinquishment policy reduces the number of relinquishment, but requires a queue and the calls' acceptance of possibly waiting in the queue. If the waiting time is not nominal, its applicability is limited to the delay insensitive services only. With the delayed relinquishment policy, however, in-progress calls use higher bandwidth than with the immediate relinquishment policy, thereby resulting in shorter transmission times. Hence, the delayed relinquishment does not necessarily cause longer call holding times or call delays, which may be an extra waiting time plus a shorter transmission time. This is one of the questions investigated in the analytical study in Chapter 6.

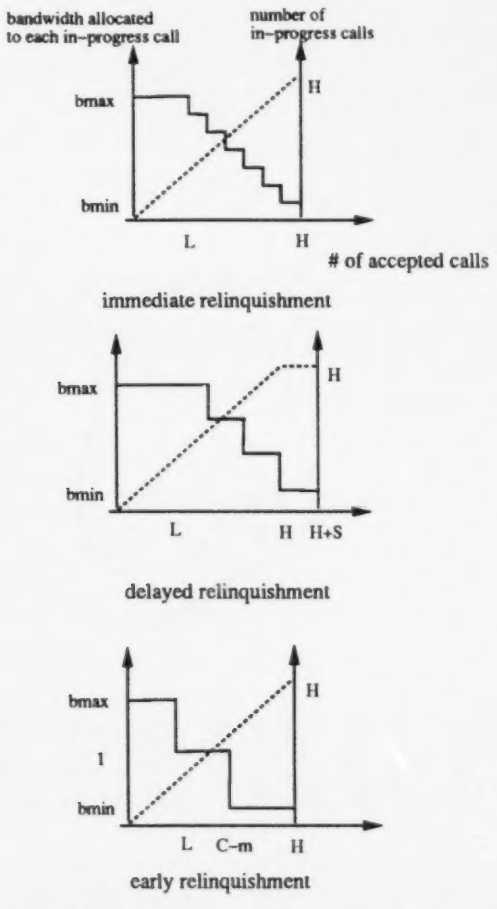

Figure 4.11: When to Relinquish: Immediate, Delayed or Early

The early relinquishment policy has been designed especially for traffic with fixed calls and flexible calls. When there are both fixed and flexible calls competing for the same bandwidth, these two types of calls can interfere with each other. The flexible calls can expand their bandwidth allocation to the maximum at every opportunity, hence leaving little bandwidth for fixed calls. On the other hand, each arrival and departure of fixed calls may disturb flexible calls through relinquishment and re-allocation.

Limits on the number of flexible cals for different bandwidth allocation choices are set in the early relinquishment policy in order to stabilize the allocation and isolate fixed calls from flexible calls. The bandwidth allocation for flexible calls is changed only when the number of in-progress flexible calls reaches one of these limits, thereby leaving some room for possible fixed call arrivals.

Departures of fixed calls will not cause bandwidth changes for flexible calls, only departures and arrivals of flexible calls themselves can trigger their bandwidth allocation change.

The granularity of bandwidth allocation is determined by these limits. For example, flexible calls are allocated a maximal bandwidth when calls are less than 25% of the capacity, and they are relinquished to the minimum bandwidth as soon as there are 75% calls, and they are given one bandwidth unit when the call number is between these two limits.

For an accepted call j and each affected in-progress call $i \in I$ (by Rule 1 and Rule 2), an optimal choice of each b_i and new b_j has to be made so that the network performance is maximized. This optimization problem is modeled as a flexible knapsack problem in the next chapter.

Chapter 5

Static Allocation

An access control can be based on either instantaneous or long term situations. When traffic is a known stochastic process, then decisions can be made based on the long term statistics. This is referred to as "dynamic control." In many situations future traffic conditions are either unknown or too complicated. In this case, access control has to be based on instantaneous performance. This is referred to as "static control." The static control problem for flexible bandwidth service and FACT is studied in this chapter. Note that static control does not mean that the control does not change as time presses. Instead, the notions "static" and "dynamic" refer to the traffic situations on which the contorl decisions are based. Static control does not know, or does not attempt to predict, the future traffic processes, while the dynamic control assumes prior knowledge of the stochastic processes of traffic streams. The static control problem for flexible bandwidth service and FACT is studied in this chapter.

Flexible calls can be served with any bandwidth within a certain range. This introduces a new dimension to the admission control and bandwidth allocation problem.

Conventionally, the admission control and bandwidth allocation decision is made for a fixed bandwidth request. The network either accepts this request and allocates that particular bandwidth to it, or rejects it. The static control problem can be modeled as a knapsack problem. The next section provides a brief overview of the knapsack problem.

The problem, however, is remarkably complicated with the new flexible bandwidth call and FACT. Because several bandwidth allocation option is now available, instead of one, all of them have to be examined before an admission control and bandwidth allocation decision can be made. A new type of knapsack problem is defined in Section 5.2. It can model the static control problem for flexible calls and many other practical resource allocation problems where more than one allocation options exist.

Section 5.3 provides several solution algorithms for flexible knapsack problems. It is then followed by results comparing the flexible knapsack to the conventional one, and the heuristic algorithm to the exact one.

5.1 Conventional Knapsack Problems

A great variety of practical problems, including the admission control and bandwidth allocation problem in communication networks, can be represented by a set of objects, each having an associated reward value and a volume value, from which a subset has to be selected such that the total reward value of the selected objects is maximized, and the total volume does not exceed some prefixed bound. These problems are generally called ***knapsack problems*** [51].

The knapsack problem commonly considered is:

$$\begin{cases} \textit{maximize} & \sum_{i=1}^{K} x_i r_i \\ \textit{subject to} & \sum_{i=1}^{K} x_i w_i \leq F \\ & 0 \leq x_i \leq b_i, 1 \leq i \leq n \end{cases}$$

where, for class i, r_i, w_i, x_i and b_i are the reward rate, volume ratio the number of objects packed in the knapsack and the number of objects available for packing, respectively.

The above problem is referred to as a general constrained single knapsack problem if $b_i < \infty$, and a general unconstrained single knapsack problem if b_i approaches ∞ [49]. The multiple knapsack problem is a generalization of the single knapsack problem where several containers are available. The knapsack problems considered here are of the single knapsack type, unless explicitly stated otherwise.

A special version of the general constrained knapsack problem commonly studied is the 0/1 knapsack problem with the x_i variables taking the value of 0 or 1 only, i.e., $b_i = 1$ [25] [30]. A generalization of the general constrained knapsack problem is a multiconstraint general knapsack problem, where an object may consume multiple resources, as opposed to having a single volume constraint [57]. The multiconstraint general knapsack problem takes the form:

$$\begin{cases} \textit{maximize} & RX \\ \textit{subject to} & WX \leq B \end{cases}$$

where W and B are constant arrays, RX is the cross product of X and R, vectors of x_i and r_i, respectively.

In communication networks the access control and bandwidth allocation problem is to find the optimal bandwidth allocation with maximum total reward. Suppose the given bandwidth is F, and b_i is the number of class i call requests for w_i bandwidth each, resulting in r_i units of reward (revenue) if accepted. Then the access control and bandwidth allocation problem can be stated as

the corresponding knapsack problem. The optimal decision is to accept x_i class i calls, and to reject the other $b_i - x_i$ call requests.

Conventional knapsack problems deal with objects of fixed volume. In the context of access control and bandwidth allocation in communication networks, only fixed bandwidth calls can be modeled by the conventional knapsack approach. The new flexible bandwidth service requires a different type of knapsack model. A new type of knapsack model – the flexible knapsack – is defined and its solutions are provided in this chapter. It can be used to model not only flexible bandwidth service but also a wide range of practical applications in other areas.

5.2 The Flexible Knapsack Problem

5.2.1 Flexible Knapsack Problem Definition

Suppose there are K different classes of objects to be packed into a knapsack of volume F. A total of b_i objects can be taken from class i. Each class i object can assume any of $n+1$ different forms, called subclasses $(i,0),(i,1),...,$ and (i,n). For simplicity, objects of subclass (i,j) will be referred to as objects (i,j). An object (i,j) occupies volume $w_{i,j}$ in the knapsack and generates reward $r_{i,j}$. The objective is to maximize the total reward of the knapsack under the knapsack capacity constraint, F, and object availability constraints, b_i's. The problem can be, thus, stated as:

$$nFKP \begin{cases} \textit{maximize} & \sum_{i=1}^{K}\sum_{j=0}^{n} r_{i,j}x_{i,j} \\ \textit{subject to} & \sum_{i=1}^{K}\sum_{j=0}^{n} w_{i,j}x_{i,j} \le F \\ & \sum_{j=1}^{n} x_{i,j} \le b_i,\ 1 \le i \le K \end{cases}$$

where integer $x_{i,j}$ is the number of objects (i,j) packed in the knapsack. The $r_{i,j}$'s, $w_{i,j}$'s and b_i's are assumed to be positive. Without loss of generality, we also assume that $b_i w_{i,j} < F$ for any i and j, and $\sum_{i=1}^{K} b_i w_{i,s} > F$, where (i,s) is the subclass at which a class i object has the largest reward to volume ratio, i.e., $r_{i,s}/w_{i,s} = \max\{r_{i,j}/w_{i,j} | 0 \le j \le n\}$.

This problem will be referred to as the general *flexible knapsack problem* (FKP) of flexibility n, or *n-flexible knapsack problem* (nFKP). The conventional knapsack has 0 flexibility, or only one subclass. Taking the bound b_i into consideration, we can divide the FKP into two categories. The FKP will be referred to as either the *unconstrained flexible knapsack problem* (UFKP), if there is no bound on any of the $x_{i,j}$'s, i.e., $b_i > \max_{0 \le j \le n}\{\lfloor F/w_{i,j} \rfloor\}$, or otherwise the *constrained flexible knapsack problem* (CFKP).

To model the optimal admission control and bandwidth allocation problem stated in the previous chapter, the classes in CFKP are the classes of calls, and the subclasses are the different choices of bandwidth allowed for this particular class. Hence, K is the number of call classes supported by the network, and b_i is number of class i calls in the network. There are $n+1$ different bandwidth op-

tions allowed for each class i, they are $w_{i,0}, w_{i,1}, ..., w_{i,n}$, yielding rewards (revenue) $r_{i,0}, r_{i,1}, ..., r_{i,n}$, respectively. The solution $x_{i,j}$ is therefore the number of class i calls to be accommodated at subclass (i,j), i.e., with bandwidth $w_{i,j}$, so that the total reward is maximized.

5.2.2 Practical Applications of the Flexible Knapsack

The flexible knapsack can find applications in many resource allocation problems. Three examples are given below.

Figure 5.12 depicts a 1FKP using sausage making as an example, where objects (different types of meat) from the same class (bucket) can be put into different subclasses (funnels of different sizes), yielding different volume requirements and rewards (different sausages), where the objective is to maximize the total reward (profit) in the knapsack (a smoke oven).

An FKP application in computing could be a data file archiving service. Files could be text or images. Each file can be compressed at different ratios. The reward could be defined as display quality when recovered from the compression. Given some files of both text and image types, an optimization decision problem arises as to which files should be compressed, and at what ratios, so that the overall quality is maximized.

Another application could arise in merchandise inventory management. Let us consider a merchant shipping supplies to a remote store from a warehouse by ship. Each kind of goods has several different packaging options providing different protections and utilization of ship cargo space. The reward for the merchant is the profit realized by selling the goods less the cost for damage, packaging, and ship rental. The optimization problem again is a flexible knapsack problem.

5.2.3 Flexible Knapsack Versus Conventional Knapsack

The conventional single constraint knapsack problem does not consider the possibility of allowing different forms of an object to be packed. The flexible knapsack proposed here models a more general situation: Objects from the same class can take different forms for packing.

The conventional knapsack problem originally gets its name from the practical situation where a hiker has to decide what food to pack in his knapsack to maximize the benefit. Now we consider a more complicated packing situation faced by a modern hiker/astronaut who has several choices of having his foods processed (e.g., smoked, canned or dehydrated), and thereby changing the volume these foods occupy and also their nutritional value (or taste reward value). To maximize the benefit, the decision now is not only which and how much food to pack, but also how much food to be processed and with what methods. The constraints now are both the knapsack's capacity and the total amount of the same foods processed differently.

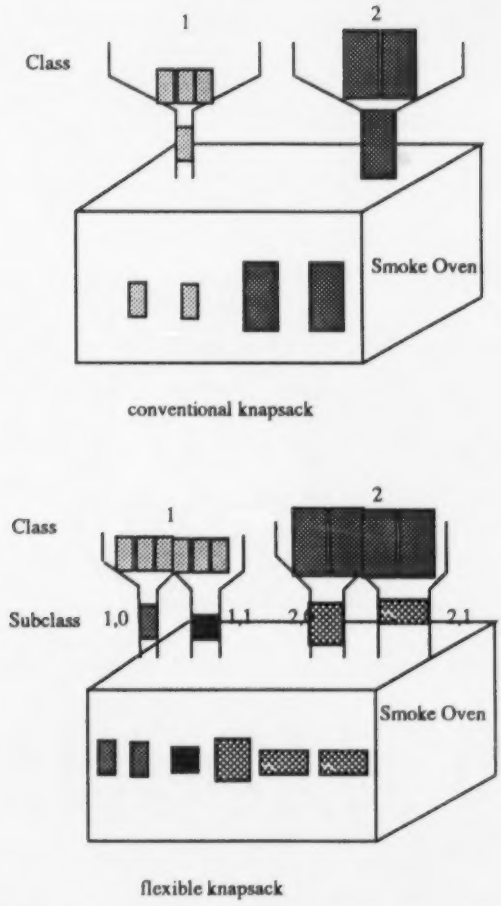

Figure 5.12: Example of a Flexible Knapsack Problem with Two Subclasses

Theorem 5.1 *The nFKP is NP-hard.*

Proof. The conventional knapsack problem, a known NP-hard problem, reduces to the 0FKP. □

The conventional multiconstraint knapsack is basically a universal blanket problem for any integer programming with a simple product-summation objective function. The flexible knapsack is a special case of the multiconstraint knapsack where the ith row of the constraint matrix W in the multiconstraint knapsack is 1 for all entries corresponding to the objects from class i, and 0 for all the rest. Note that the constraint array W can be of arbitrary form in a multiconstraint knapsack. Hence, its solution has to cope with any form of W. Clearly this general model is inefficient when only a special form of W is needed as in FKP. Hence, the importance of FKP warrants the need for a specific study. The relationship between the multiconstraint knapsack problem and FKP is

analogous to that of integer programming and the multiconstraint knapsack, in the sense that all multiconstraint knapsack problems can be solved with integer programming, but they themselves are sufficiently important to be studied separately from the other general integer programming problems. Figure 5.13 illustrates the relationship between the flexible knapsack and the conventional ones.

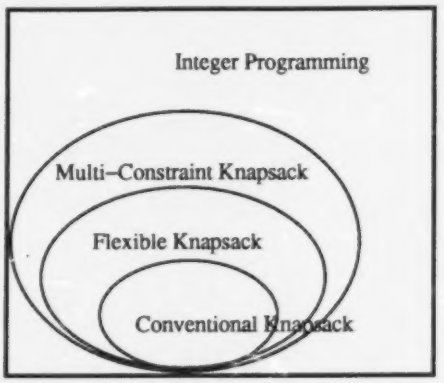

Figure 5.13: Relation Between Flexible and Conventional Knapsacks

5.2.4 Modeling the Flexible Access Control Problem

The flexible knapsack model proposed here arises from the problem of optimizing reward or revenue for communication networks. If the objective is to maximize revenue, the reward $r_{i,j}$ in the FKP model clearly is the telecommunication tariff less the cost for providing a type (i,j) call service. The volume $w_{i,j}$ is the bandwidth needed by a type (i,j) call.

The flexible knapsack model can also be used to study other performance measurements for access control. If we are to maximize channel utilization, then $r_{i,j} = w_{i,j}$ will be the bandwidth needed by a type (i,j) call.

To use the FKP model for cost minimization, the reward $r_{i,j}$ will represent the cost of providing the service to a type (i,j) call, and maximization in the algorithms for the FKP is replaced by minimization.

Let the solution to the FKP be $X^* = \{x^*_{i,j} | i \in [1,K], j \in [0,n]\}$. The network will then accept $\sum_{j=0}^{n} x^*_{i,j}$ of the b_i requests, allocate $w_{i,j}$ channels to $x^*_{i,j}$ calls, and reject the rest of the call requests. The blocking rate for a type i call is thus $1 - \sum_{j=0}^{n} x^*_{i,j}/b_i$.

Once we have an effective and efficient solution to FKP, most of the network performance measures can be obtained.

5.3 Solutions to Flexible Knapsack Problems

The exact solution approaches to UFKP and CFKP are designed in the first two subsections, respectively. A heuristic solution is provided for CFKP in the last subsection.

5.3.1 Solution to the Unconstrained Flexible Knapsack Problem

The nUFKP can be converted into a conventional general unconstrained knapsack problem (GUKP) of $(n+1)K$ variables:

$$GUKP \begin{cases} \textit{maximize} & \sum_{i=1}^{(n+1)K} \overline{x}_i \, \overline{r}_i \\ \textit{subject to to} & \sum_{i=1}^{(n+1)K} \overline{x}_i \, \overline{w}_i \leq F; \end{cases}$$

with the variable substitutions of index (i,j) in nUFKP by index $v = (i-1)(n+1)+j$ in GUKP, i.e.,

$$\begin{aligned} \overline{x}_v &= x_{i,j} \\ \overline{r}_v &= r_{i,j} \\ \overline{w}_v &= w_{i,j} \\ v &= (i-1)(n+1)+j \end{aligned}$$

The corresponding GUKP can be solved either directly by using existing algorithms such as dynamic programming [25] [30], branch and bound [49] [50], or by converting it into a conventional knapsack problem of $(\sum_{i=1}^{K} \sum_{j=0}^{n} \log_2 F/w_{i,j})$ variables that take value 0 or 1 only, and then solving the conventional 0/1 knapsack problem with the known algorithms such as those summarized by Martello and Toth [51].

5.3.2 Solution to Constrained Flexible Knapsack Problems

For the sake of simplicity, we only use the 1CFKP in the remaining part of this thesis except where explicitly stated otherwise. Thus only two subclasses are considered for each class. Nevertheless, the results can be extended to solve the general nCFKP for any integer n.

In contrast to the UFKP, the CFKP cannot be converted to the conventional knapsack problem, because the variables $x_{i,j}$ in the CFKP are not independent of each other within each class i. All variables in the conventional knapsack problem, however, must be independent of each other.

A. Enumeration

The most straight forward solution method is enumerating all feasible solutions, i.e., all $x_{i,j}$ satisfying the constraints. The enumeration algorithm for 1CFKP, $\mathbf{em}(\cdot)$, employs a recursive function that returns the maximal values for objects of classes $j \geq i$, given i and the free volume E. The maximum reward for 1CFKP is then $\mathbf{em}(1, F)$. The function $\mathbf{em}(\cdot)$ can be stated as:

```
Function em(i, E)
        if i > K or E ≤ 0
          return(0);
        maxi = em(i + 1, E);
        for s = 0 to min{b_i, E/w_{i,0}}
            for t = 0 to min{b_i − s, (E − sw_{i,0})/w_{i,1}}
                if sr_{i,0} + tr_{i,1} + em(i + 1, E − sw_{i,0} − tw_{i,1}) > maxi
                  maxi = sr_{i,0} + tr_{i,1} + em(i + 1, E − sw_{i,0} − tw_{i,1})
        return(maxi);
```

Clearly, the time complexity of the enumeration algorithm is $O(\prod_{i=1}^{K} b_i^2)$. It becomes $O(F^{2K})$ when $b_i = O(F)$. The enumeration is too time consuming to be of practical use. Nevertheless, it provides the most reliable solution because of its simplicity. In this study, it was used to verify all the new algorithms by comparing their results for each data sample used.

B. Dynamic Programming

Let $R_i(E)$ be the maximum reward for a subproblem of 1CFKP where only objects from classes i and higher are to be packed into the available volume E. Thus $R_i(E)$ is the optimum solution of the following problem:

$$CFKP-DP \begin{cases} \textit{maximize} & \sum_{k=i}^{K}(x_{k,0}r_{k,0} + x_{k,1}r_{k,1}) \\ \textit{subject to to} & \sum_{k=i}^{K}(x_{k,0}w_{k,0} + x_{k,1}w_{k,1}) \le E \\ & x_{k,0} + x_{k,1} \le b_k,\ i \le k \le K \end{cases}$$

Hence, the solution of CFKP-DP contains the solution of the CFKP as a special case with $i = 1$ and $E = F$. In other words, solving 1CFKP amounts to finding $R_1(F)$.

The function $R_i(E)$ is clearly decomposable and also monotonically nondecreasing relative to each $x_{k,j}$. Therefore, based on the optimality principle [52], the equations for obtaining $R_1(F)$ recursively are

$R_i(E) = \max\{x_{i,0}r_{i,0} + x_{i,1}r_{i,1} + R_{i+1}(E') | x_{i,0} + x_{i,1} \le b_i,\ x_{i,0}w_{i,0} + x_{i,1}w_{i,1} \le E\}$,

$R_{K+1}(E) = 0,\ E > 0$, and

$R_i(E) = 0,\ E \le 0$, where $E' = E - x_{i,0}w_{i,0} - x_{i,1}w_{i,1}$ is the free volume left after packing $x_{i,0}$ objects at subclass 0 and $x_{i,1}$ objects at subclass 1.

From these recursive equations, a (backward) dynamic programming algorithm for 1CFKP is derived as shown in the Algorithm DP1, where $f_i(E, s)$ stores the value of $R_i(E)$ when $x_{i,0} \le s$ and $x_{i,1}$ is fixed at 0.

In the algorithm, the maximum reward $R_1(F)$ is achieved when $x_{i,j} = x_{i,j}^*$. Variable $f_i(E, s)$ stores the value of $R_i(E)$ when $x_{i,0} \le s$ and $x_{i,1} = 0$. The set $trace(E, i)$ keeps track of the optimal value of $\{x_{i,0}, x_{i,1}\}$, given a free volume E.

Algorithm: DP1

1. Stage K:
   i = K;
   For E=0 to F
   $d = \min(b_i, E/w_{i,0})$;
   For s=0 to d
   $f_i(E,s) = s\,r_{i,0}$
   $optx_0(E,s) = s$;
   For s=d+1 to b_i
   $f_i(E,s) = d\,r_{i,0}$
   $optx_0(E,s) = d$;
   $d = \min(b_i, E/w_{i,1})$;
   p $R_i(E) = f_i(E,b_i)$;
   $trace(E,i) = \{optx_0(E,b_i), 0\}$;
   For t=1 to d
   if $R_i(E) < tr_{i,1} + f_i(E - tw_{i,1}, b_i - t)$
   then $R_i = tr_{i,1} + f_i(E - tw_{i,1}, b_i - t)$
   $trace(E,i) = \{optx_0(E - tw_{i,1}, b_i - t), t\}$;
2. Stage 1,..., 2:
   For i=1 to 2
   For E=0 to F
   $d = \min(b_i, E/w_{i,0})$
   $f_i(E,0) = R_{i+1}(E)$;
   For s=0 to d
   if $(sr_{i,0} + R_{i+1}(E - sw_{i,0}), f_i(E, s-1))$;
   then $f_i(E,s) = sr_{i,0} + R_{i+1}(E - sw_{i,0})$;
   $optx_0(E,s) = s$;
   else $f_i E, s = f_i(E, s-1)$;
   $optx_0(E,s) = 0$
   For s=d+1 to b_i
   $f_i(E,s) = f_i(E, s-1)$;
   $optx_0(E,s) = optx_0(E,d)$;
   $d = \min(b_i, E/w_{i,1})$;
   $R_i(E) = f_i(E,b_i)$;
   $trace(E,i) = \{optx_0(E,b_i), 0\}$;
   For t=1 to d
   if $R_i(E) < tr_{i,1} + f_i(E - tw_{i,1}, b_i - t)$
   then $tr_{i,1} + f_i(E - tw_{i,1}, b_i - t)$;
   $trace(E,i) = \{optx_0(E - tw_{i,1}, b_i - t), t\}$;
3. Stage 1:
   i=1;
   For E=0 to F
   $d = \min(b_i, E/w_{i,0})$;
   $f_i(E,0) = R_{i+1}(E)$;
   For s=0 to d
   if $(sr_{i,0} + R_{i+1}(E - sw_{i,0}), f_i(E, s-1))$
   then $f_i(E,s) = sr_{i,0} + R_{s+1}(E - sw_{i,0})$;
   $optx_0(E,s) = s$;
   else $f_i E, s = f_i(E, s-1)$;
   $optx_0(E,s) = 0$
   For s=d+1 to b_i

$f_i(E,s) = f_i(E,s-1)$;
$optx_0(E,s) = optx_0(E,d)$
$d = \min(b_i, E/w_{i,1})$;
$R_i(F) = f_i(F,b_i)$;
$x^*_{i,1} = 0$;
For t=1 to d
if $R_i(F) < tr_{i,1} + f_i(F - tw_{i,1}, b_i - t)$
then $tr_{i,1} + f_i(F - tw_{i,1}, b_i - t)$;
$trace(F,i) = \{optx_0(F - tw_{i,1}, b_i - t), t\}$;
$i = 1$, $E = F$;
while $(i <= K)$ and $(E > 0)$
$\{x^*_{i,0}, x^*_{i,1}\} = trace(E,i)$
$E = E - x^*_{i,1}w_{i,1} - x^*_{i,0}w_{i,0}$
$i = i + 1$;

Theorem 5.2 *The time complexity of the algorithm DP1 is* $O(KF\sum_{i=1}^{K} b_i)$.

Proof. . At each stage i, $O(Fb_i)$ computations are required. Thus, the total computation cost for K stages is $O(KF\sum_{i=1}^{K} b_i)$. □

The algorithm DP1 for 1CFKP can be extended easily to nCFKP. Clearly, we have

Corollary 5.1 *The time complexity of the dynamic programming algorithm for general nCFKP derived from DP1 is* $O(nKF\sum_{i=1}^{K} b_i)$.

In a telecommunication network, the number of different call types K and the number of possible forms for each type of service n could be relatively small. The number of free channels F and the number of requests b_i's might be moderate for a narrow-band digital pipe, and quite large for a wide-band digital pipe. Hence, the computation cost could be prohibitively high for some situations making the dynamic programming approach unsuitable for online decisions.

C. Branch and Bound Algorithm for General 1CFKP

C.1. 0/1 Flexible Knapsack

A flexible knapsack is called 0/1 flexible knapsack when each $b_i = 1$ in CFKP. Variable $x_{i,j}$ in 0/1 CFKP can, thus, only take on the value 0 or 1 for any i and j. This terminology parallels that of the conventional 0/1 knapsack problem.

Assume that all objects are sorted so that $r_{i,0}/w_{i,0} \geq r_{i,1}/w_{i,1}$, and $r_{i,0}/w_{i,0} \geq r_{i+1,0}/w_{i+1,0}$. We also have $r_{i,0} \neq r_{i,1}$, $w_{i,0} \neq w_{i,1}$ and $\sum_{i=1}^{K} b_i w_{i,0} > F$ as assumed for the general FKP in its definition.

Theorem 5.3 *Let*

$$\begin{aligned} l &= \max\{j \mid \sum_{i=1}^{j} w_{i,0} \leq F\}; \\ R &= \sum_{i=1}^{l} r_{i,0} \\ E &= F - \sum_{i=1}^{l} w_{i,0} \\ P_l &= \max\{(r_{i,1} - r_{i,0})/(w_{i,1} - w_{i,0}) > 0 | 1 \leq i \leq l\} \end{aligned}$$

Then $R_u = R + \max\{\lfloor Er_{l+1,0}/w_{l+1,0} \rfloor, \lfloor EP_l \rfloor\}$ *is an upper bound on the total reward of the 0/1 1CFKP.*

Proof. The reward R considered here is obtained by consecutively packing objects at subclass 0, that is, the "best" subclass, until a class $l+1$ object is reached which cannot be packed into at subclass 0.

The optimal solution can now be obtained by either inserting or not inserting any objects of class $k > l$, and/or by substituting different forms for the objects already in the knapsack. Thus, altogether there are three possible situations as shown in Table 5.3.

Table 5.3: Cases for Obtaining the Optimum

Objects Packed	No More Packing	More Packing
no sub.	–	Case 1
substituting	Case 2	Case 3

The total reward under Case 1 cannot exceed $R + \lfloor Er_{l+1,0}/w_{l+1,0} \rfloor$, because for any new object packed, its reward volume ratio is no more than that of object $(l+1, 0)$ as a result of the sorting. The total reward is bounded by $R + \lfloor EP_l \rfloor$ in Case 2 by the definition of P_l. In Case 3, a bound R_u is obtained by noticing that the improvement from R is limited by the reward volume ratio $\max\{r_{l+1,0}/w_{l+1,0}, P_l\}$.

Thus, R_u is a valid upper bound for the 0/1 1CFKP. □

Based on this theorem, a branch and bound algorithm for 0/1 1CFKP is described below, where the *projected gain* is obtained by the same approach as in the proof, the *recorded maximum* is the maximum reward obtained so far, and the *realized gain* is the total reward achieve with current packing.

Algorithm: BB0

1. Initialization
   sort all items so that $r_{i,0}/w_{i,0} \geq r_{i,1}/w_{i,1}$, and $r_{i,0}/w_{i,0} \geq r_{i+1,0}/w_{i+1,0}$;
   calculate R_u; set $i = 1$.
2. Probing Packing
   repeat packing consecutively from i
   until $(E < w_{i,0})$ or $(i > K)$
   if the projected gain for this packing is less than the recorded maximum
   then give up this probing and go to Backtracking
   else put the probing objects into knapsack
   if $i \leq K$
   then find j such that $j > i$ and $w_j \leq E$
   if such j exits then $i = j$, repeat probing packing
3. Updating
   if the realized gain of this packing is better than the recorded maximum
   then update optimal record
   if the realized gain of this packing equals R_u
   then stop; the optimal solution is given by the optimal record
   else $i = K$
4. Backtracking
   if the knapsack is empty, then stop; the optimal solution is given by the optimal record.
   find an object from class k which is the closest to i in the knapsack
   if this class k object is a $(k, 0)$
   then project the maximal gain by using k instead of l
   if projected gain is greater than the recorded maximum
   then go to Second Choice
   remove object-k
   if projected gain is greater than the recorded maximum
   then go to probing packing
   else $i = k$ and repeat backtracking
5. Second Choice
   change the current class k object to (k,1),
   go to probing packing

The numerical results are presented in the next section, and are compared to other algorithms.

C.2. General 1CFKP

The bound b_i is not necessarily 1 in a general 1CFKP. Assume again that all objects are sorted such that $r_{i,0}/w_{i,0} \geq r_{i,1}/w_{i,1}$ and $r_{i,0}/w_{i,0} \geq r_{i+1,0}/w_{i+1,0}$.

Theorem 5.4 *Let*

$$l \;=\; \max\{j | \sum_{i=1}^{j} b_i r_{i,0} \leq F\};$$

$$
\begin{aligned}
R &= \sum_{i=1}^{l} b_i r_{i,0} + \lfloor (\sum_{i=1}^{l} b_i w_{i,0})/w_{l+1,0} \rfloor r_{l+1,0}; \\
E &= F - \sum_{i=1}^{l} b_i w_{i,0} - \lfloor (\sum_{i=1}^{l} b_i w_{i,0})/w_{l+1,0} \rfloor w_{l+1,0}; \\
P_l &= \max\{(r_{i,1} - r_{i,0})/(w_{i,1} - w_{i,0}) > 0 | 1 \le i \le l\}
\end{aligned}
$$

Then $R_u = R + \lfloor E \max\{r_{l+1,0}/w_{l+1,0}, P_l\} \rfloor$ *is an upper bound for the general 1CFKP.*

Proof. Similar to that of Theorem 5.3, hence omitted. □

Based on this theorem, we have the following algorithm, using $\bullet$ as an index denoting all possible values of that index.

Algorithm BB1

1 Initialization
  Sort all items such that $r_{i,0}/w_{i,0} \ge r_{i,1}/w_{i,1}$ and $r_{i,0}/w_{i,0} \ge r_{i+1,0}/w_{i+1,0}$.
  compute U and P_i.
  set $r_{K+1,\bullet} = 0; w_{K+1,\bullet} = 1, b_{K+1} = F + 1$
  $Max = 0, x_{\bullet,\bullet} = 0; E = R = 0, i = 1$
2. Constructing Initial Solution R
  repeat
    $E = E - b_i w_{i,0}, R = R + b_i r_{i,0}, x_{i,0} = b_{i,0}, x_{i,1} = 0,$
    $i = i + 1, s = \lfloor E/w_{i,0} \rfloor$.
  until $(s < b_i)$
  $E = E - s w_{i,0}, R = R + s r_{i,0}, x_{i,0} = s, x_{i,1} = 0.$
3. Recursive Improvement
  If $E = 0$
    then
      the optimal solution has been found by the initial solution
    else
      call the subroutine **recur-bb**$(1, E, R)$

The function *recur-bb(from, cap, gained)* is the key of the algorithm. It utilizes a depth-first branch and bound search tree to find an optimum solution. When objects from class less than *from* have been determined, then volume E is left and reward *cap* is achieved. The function returns the boolean value true if the search is successful, and also updates the optimal record Max when necessary. At each node of the search tree, this function selects a not yet checked class, say the ith, to generate descendant nodes by assigning $x_{i,0}$ and $x_{i,1}$ to all possible values. The search then moves to the node having the maximum reward, i.e, with $x_{i,0} = b_i$. At each move, the upper bound is tested for future search. If the upper bound is lower than the current optimum solution, then the search should not proceed from this node and backtracking is called for. At the backtracking stage, the current node is abandoned, and its siblings are tested.

Function recur-bb(*from*, *cap*, *gained*)

1. Start:
   $i = from, E = cap, R = gained$;
2. Packing:
   while $E \geq b_i w_{i,0}$ and $i \leq K$
   $x_{i,0} = b_i; x_{i,1} = 0;$
   $E = E - b_i w_{i,0}; R = R + b_i r_{i,0};$
   if $(i <= K)$ and $(E/w_{i,0} > 0)$
   $x_{i,0} = \lfloor E/w_{i,0} \rfloor; x_{i,1} = 0;$
   $E = E - x_{i,0} w_{i,0}; R = R + x_{i,0} r_{i,0};$
   if $(E > 0)$ and $(Max > R + E r_{i+1,0}/w_{i+1,0})$
   then go to Backtracking;
   else if $i < K$
   then i++; go to Packing;
   $i = from$;
   while $(E > 0)$ and $(i <= K)$
   $tk = \min(b_i - x_{i,0} - x_{i,1}, E/w_{i,1});$
   if $tk > 0$
   $E- = tk w_{i,1};$
   $R+ = tk r_{i,1};$
   $x_{i,1}+ = tk;$
   i++;
3. Updating
   if $R > Max$
   Max = R
   copy x to X
   i =K;
4. Backtracking
   k=i;
   While $x_{k,0} = 0$ and $x_{i,1} = 0$ and $k \geq from$
   ;
   if $(k < from)$
   go to found;
   save $x_{k,\bullet}, x_{k+1,\bullet}, \ldots, x_{K,\bullet}$
   try removal or change of $x_{k,0}$ by calling **change**(k, E, R)
   try removal of $x_{k,1}$ by calling **remove1**(k, E, R)
   remove all $x_{k,0}, x_{k,1}$
5. Top-off
   for $k = from$ to $k - 1$
   pack $x_{k,0}$ if possible
   for $k = from$ to $k - 1$
   pack $x_{k,1}$ if possible
   if $R > Max$
   then update the optimum record
   if $Max < R + E \bullet r_{k,0}/w_{k,0}$
   then $i = k$; and go to Packing;
   else $i = k - 1$; and go to Backtracking
6. Found
   if $R > Max$
   then update the optimum record
   if an improvement is made by this call

```
            then return(true)
            else return(false)
    Function change(k, cap, gain)
R = gain; E = cap;
while there is still object (k, 0)'s
        remove one object (k, 0),
        if (Max < R + E r_{k+1,0}/w_{k+1,0})
          then call recur-bb(k + 1, E, R);
        up = max{b_k - x_{k,0} - x_{k,1}, E/w_{k,1}}
        for ch = 1 to up
            add one object (k, 1)
            if new R > Max
              then update the optimal record
            if (Max < R + E r_{k+1,0}/w_{k+1,0})
              then call recur-bb(k+1, E, R)
    Function remove1(k, E, gained)
where there is still object (k, 1) left
        remove one object (k, 1)
        if (Max < gain + E r_{k+1,0}/w_{k+1,0})
          then call recur-bb(k + 1, E, gain)
```

5.3.3 Heuristic Solution to Constrained Flexible Knapsack Problems

To reduce the computational complexity, a heuristic solution is developed. The CFKP is first simplified by considering only the "best" subclass for each class. The best subclass considered is the one with the maximum reward to volume ratio among all subclasses from the same class, i.e., the subclass (i, s) for a given i ($r_{i,s}/w_{i,s} = \max\{r_{i,j}/w_{i,j} \mid 0 \le j \le n\}$), the $r_{i,s}$ and $w_{i,s}$ respectively. The heuristic solution is then obtained by solving the following problem:

$$AP-CFKP \begin{cases} maximize & \sum_{i=1}^{K} \overline{x_i} \overline{r_i} \\ subject\,to & \sum_{i=1}^{K} \overline{x_i} \overline{w_i} \le F \\ & \overline{x_i} \le b_i \; 1 \le i \le K \end{cases}$$

The problem AP-CFKP is a conventional general unconstrained knapsack [51] or a knapsack with multiple choice constraints [57]. Its solution has been well studied and documented. The heuristic solution algorithm is stated below.

Algorithm HEUn

1. Reduction
   For $i = 1$ to K
   find the s such that $r_{i,s}/w_{i,s} = \max\{r_{i,j}/w_{i,j} \mid 0 \le j \le n\}$;
   let $\overline{r}_i = r_{i,s}, \overline{w}_i = w_{i,s}, l_i = s$;
2. Solving AP-CFKP
   call a known conventional knapsack algorithm to solve AP-CFKP;
   Let the reward obtained be R;
3. Construct Solution
   $x_{i,j} = 0,\ 1 \le i \le K, 0 \le j$'$K, j \ne l_i$

$x_{l_i} = \overline{x}_i$, $1 \leq i \leq K$ and the total reward is R.

The complexity of this heuristic algorithm is primarily the complexity of the conventional knapsack algorithm used. Clearly, the solution obtained with this heuristic algorithm may not be the optimum solution to the nCFKP, because of neglecting the subclasses other than the "best" ones. However, as shown with the results in the next chapter, this heuristic algorithm can reduce time complexity of the flexible knapsack problem tremendously, yet provides solutions very close to the optimum in most cases.

5.4 Numerical Results

We implemented both the exact solution and the heuristic solution with programs written in C running on a MIPS RISC/OS 4 system operating at 12 million instructions per second. The algorithm used for the conventional knapsack problem is one of the fastest algorithms developed so far, namely Martello and Toth's algorithm [49].

Each program was run with different combinations of capacity F and number of classes K. For each combination of F and K, a sequence of 1,000 random samples were used as input to obtain the results. Each sample is a set of values of $w_{i,j}, r_{i,j}$ and b_i, all of which are generated randomly from uniform distributions under the following conditions.

- $1 \leq w_{i,0} \leq \lfloor F/K \rfloor$, $w_{i,1} = w_{i-1,0}$ and $w_{1,1} = 1$;
- $1 \leq r_{i,0} \leq \lfloor F/K \rfloor$,
- $1 \leq r_{i,1} \leq r_{i,0}$,
- $\sum_{i=1}^{K} b_i w_{i,0} > F$, and $b_i \leq \max\{\lfloor F/w_{i,j} \rfloor\}$ after the $r_{i,j}$'s and $w_{i,j}$'s are sorted.

The capacity F changes from 20 to 200 with increments of 20. The number of service classes K ranges from 3 to 10 incremented by 1. As stated before, only 1CFKP is considered, hence, the number of subclasses is 2.

5.4.1 FACT Versus Conventional Scheme

First, we investigate the improvement of FACT over the conventional scheme in terms of total reward achieved. With FACT, three alternative actions are assessed for each object: rejection, acceptance at subclass 0, and acceptance at subclass 1. For each object, the conventional scheme considers only

Table 5.4: Comparison of Average Total Reward: FACT vs. Conventional

K	F	$E(R_f)$	$E(R_c)$	$E(\Delta R)(\%)$	$\max\{\Delta R\}(\%)$
3	20	145.247	145.134	0.078	13.5
	40	507.767	507.393	0.074	32.6
	60	1086.402	1085.601	0.074	28.1
	80	1902.163	1901.442	0.038	52.9
	100	2729.831	2728.236	0.058	40.3
	120	3972.148	3970.127	0.051	34.0
	140	5330.640	5329.530	0.021	11.7
	160	7177.217	7175.180	0.028	39.5
	180	8645.363	8641.520	0.044	36.3
	200	10857.013	10852.528	0.041	60.0
6	20	132.054	132.047	0.005	2.9
	40	496.996	496.958	0.008	2.9
	60	1042.914	1042.795	0.011	11.8
	80	1803.206	1803.063	0.008	10.3
	100	2814.800	2814.643	0.006	24.7
	120	4009.239	4008.458	0.019	22.1
	140	5417.916	5416.667	0.023	23.7
	160	7099.702	7099.321	0.005	3.1
	180	8647.428	8646.999	0.005	5.6
	200	11076.619	11075.730	0.008	34.4
10	20	128.669	128.669	0.000	0.0
	40	479.565	479.565	0.000	0.0
	60	1074.266	1074.258	0.001	1.0
	80	2092.749	2092.749	0.000	0.0
	100	2676.970	2676.921	0.002	3.2
	120	4011.455	4011.350	0.003	8.4
	140	5465.871	5465.808	0.001	0.5
	160	6882.306	6882.093	0.003	0.5
	180	8713.349	8713.173	0.002	1.0
	200	10888.785	10888.613	0.002	7.3

one form of acceptance which is assumed in our study as the "best" subclass, i.e., the one with the largest reward volume ratio among all subclasses.

Rewards are gathered for FACT and the conventional scheme for over 1,000 random samples described above.

The results are tabulated in Table 5.4 for various values (20-200) of the capacity F and for K=3, 6 and 10 classes. The third ($E(R_f)$) and fourth ($E(R_c)$) columns are the average total rewards (for the 1,000 runs) with FACT and the conventional scheme, respectively. The last two columns are deviation measures of the reward achieved with the conventional scheme over that of FACT ($\Delta R = (R_f - R_c)/R_f$).

For all cases considered ($F \leq 200$, $K \leq 10$), FACT always obtained rewards more than, or equal to, the rewards achieved by the conventional scheme. The average difference between the conventional scheme and FACT is less than 0.08%. The largest difference was up to 60% for small K (3). For large K, however, the difference does not exceed 10% over all cases.

To explain this variation in behavior over K, one may note that $1/K$ is proportional to the ratio of the average volume of objects to the total knapsack capacity, namely, $(1 + F/K)/2$ to F. When this ratio is small (K is large), an object likely occupies only a small fraction of the knapsack. Hence, the knapsack's total reward is relatively stable to the adjustment of objects. Therefore, the impact of changing subclasses is not as great as when the above ratio is high (K is small).

The reward differences between the conventional scheme and FACT depend on the choices of $r_{i,j}$ and $w_{i,j}$. They tend to become large with large volume difference, $w_{i,0} - wi, 1$, because an optimal choice of subclass makes a bigger difference. In addition, the reward differences are functions of the reward rate distribution. The particular choices of these distributions are considered to contribute to the small difference in average rewards between the conventional and the FACT scheme.

5.4.2 Exact Solution Versus Heuristic Solution

Recalling the heuristic algorithm, one may notice that the heuristic algorithm always obtains the same solutions as the above conventional solution. Therefore, the performance results mentioned above also apply to the heuristic algorithm. That is, the heuristic solutions are very close to the optimum, 0.08% on average for all cases studied. For large K, the heuristic solution is at most 10% less than the optimum and for small K it can be up to 60% less than the optimum.

In Figure 5.14 the computation times are compared for obtaining the exact solution and for that of the heuristic solution. The solid lines in Figure 5.14 represent the average computation time of obtaining the exact FACT solution with various number of classes K versus different capacity F. It is evident that the average computation time for FACT is of the order F^2. The computation time also increases slightly when K increases.

For larger F and K, the computation time for FACT could be too long to be suitable for real-time decisions. Thus the heuristic method discussed in Chapter 4 could be used to reduce the computation time. The dotted line close to the computation time axis in Figure 5.14 indicates the average time of obtaining the heuristic solution for various K's. The results show clearly that the heuristic solution is tremendously faster than the exact solution. In most cases, the execution time for the heuristic algorithm was in the neighborhood of 1 millisecond. This, combined with the results in Table 5.4, indicates that for large F and K the heuristic algorithm can be used as a very close and fast approximation to the flexible knapsack.

We also studied the utilization of the knapsack with the exact solution approach (U_{ex}) versus

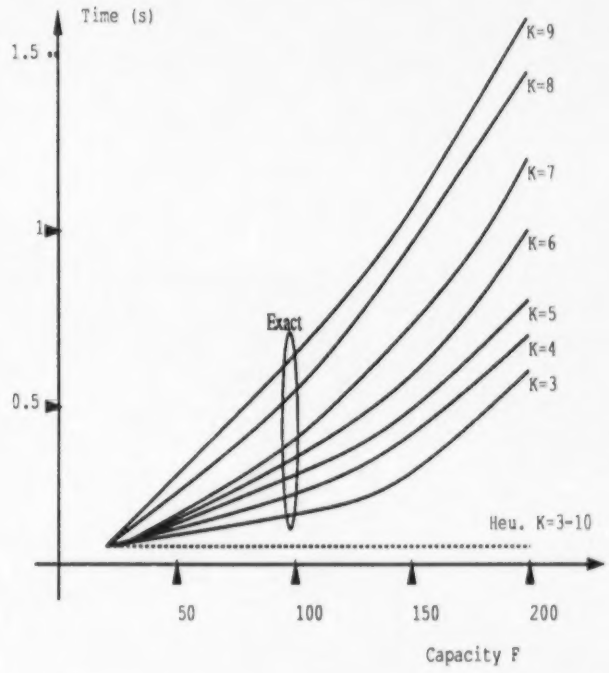

Figure 5.14: Average Computation Time of Exact vs. Heuristic Solutions

the heuristic approach (U_{heu}) (see Table 5.5). The results show that the knapsack utilization with the two methods is almost the same. Notice that the total reward is to be maximized. Hence, the utilization is not necessarily at a maximum when the reward is optimal. It is quite possible that a sub-optimal solution for the reward may occasionally yield better utilization than the optimum solution. This explains the negative items in Table 5.5.

Utilizations in Table 5.5 also show similar behavior with regard to K as did the rewards in Table 5.4. The behavior of utilization can be interpreted similarly to that of rewards. Hence, it can also be said, from the utilization viewpoint, that the FACT solution is more beneficial when the ratio of average volume of objects to the total capacity is as compared to when this ratio is large.

5.5 Summary

A special version of the knapsack problem, the flexible knapsack problem, is defined for the situation where for the same class of objects several packing alternatives (forms) are possible. This chapter discusses some practical applications of the flexible knapsack, including network access control and

Table 5.5: Average Utilization Comparison: Exact vs. Heuristic Solutions

K	F	$E(U_{ex})$	$E(U_{heu})$	$E(\Delta R)$ (%)	max$\{\Delta R\}$(%)
3	20	0.983	0.975	0.809	20.0
	40	0.972	0.962	0.957	25.6
	60	0.967	0.956	1.100	30.0
	80	0.964	0.954	1.006	23.7
	100	0.965	0.955	0.999	26.3
	120	0.966	0.953	1.318	30.8
	140	0.963	0.954	0.982	26.1
	160	0.963	0.952	1.140	27.4
	180	0.963	0.952	1.218	26.4
	200	0.963	0.954	0.914	25.7
6	20	0.999	0.999	0.035	5.0
	40	0.998	0.997	0.093	5.0
	60	0.995	0.993	0.147	6.8
	80	0.993	0.992	0.146	11.2
	100	0.996	0.996	0.008	1.0
	120	0.992	0.991	0.168	10.8
	140	0.993	0.991	0.170	8.6
	160	0.993	0.991	0.167	9.3
	180	0.992	0.991	0.185	15.0
	200	0.992	0.991	0.151	9.0
10	20	1.000	1.000	0.000	0.0
	40	1.000	1.000	0.000	0.0
	60	1.000	1.000	0.008	3.3
	80	1.000	1.000	0.000	0.0
	100	0.999	0.999	0.028	4.0
	120	0.999	0.999	0.009	0.8
	140	0.998	0.998	0.023	2.8
	160	0.990	0.989	0.119	2.5
	180	0.998	0.997	0.027	3.3
	200	0.998	0.998	0.031	3.5

bandwidth allocation, data file compression and merchandise management. The solution to the flexible knapsack problem is investigated. Algorithms DP1 and BB1 were developed with dynamic programming and branch and bound methods, respectively. The main advantage of DP1 is its simplicity. It can be used for off-line decisions, or for online ones when the knapsack is relatively small. It is, however, not suitable for real time applications of large network size. Performance experiments show that the algorithm with the branch and bound method is quite fast even for large knapsacks. The empirical results suggest that, given a number of classes K, the computation time of the branch and bound algorithm is in the order of F^2.

A heuristic algorithm for solving the flexible knapsack problem is also developed in this chapter. The performance study shows that it is tremendously faster than the exact solution algorithms.

Moreover, the heuristic solutions, on average, are very close to the optimum in most cases, especially when the capacity to average object volume ratio is small. Hence, it can be used as a very efficient approximation for large flexible knapsack problems where this ratio is small.

Coming back to the B-ISDN telecommunication example, the optimization objective considered is total revenue. The empirical studies indicate that FACT always outperforms the conventional fixed allocation scheme. The advantage of FACT is particularly evident when the average bandwidth requirement of all calls is relative large compared to the total network bandwidth capacity; in other words, when most calls require a large number of bandwidth units. On the other hand, when the average bandwidth requirement of all calls is very small, compared to the network capacity, then the performance difference between the new scheme and the old schemes is not significant. Hence, FACT is more beneficial to broadband networks which primarily serve high-bandwidth calls than to narrowband networks whose traffic consists mainly of single channel voice calls.

When the network bandwidth capacity F increases, the computation time for finding the exact solution with FACT increases in the order of F^2. Hence, for broadband access control with large F, the heuristic algorithm proposed in this chapter can provide efficient and effective solutions. The heuristic algorithm is significantly faster than the exact algorithms. The heuristic solutions are also very close to the optimum, for all traffic mixes, especially when the traffic contains a considerable amount of narrowband calls.

Hence, FACT may be used best for broadband networks. When carried traffic consists of mostly broadband calls, the exact solutions can be obtained satisfactorily with the algorithms presented in this chapter. When computation time is the major concern, the proposed heuristic algorithm can be used to provide efficient and effective approximate solutions.

Chapter 6

Dynamic Allocation: Analytical Performance Study

6.1 Overview

Static control studied in Chapter 4 deals with the current traffic situation, whereas dynamic control deals with the statistical performance over a long period of time. No consideration is given in static control to future traffic changes. Dynamic control, on the other hand, considers traffic as a stochastic process, that is, each call and burst arrive and depart randomly. Dynamic control also differs from static allocation in its requirement of some prior knowledge or assumptions of the traffic arrival and departure characteristics. Dynamic control consists of two parts: dynamic bandwidth allocation and dynamic access control; they are studied in this chapter and the next chapter, respectively.

The objectives of this chapter are to provide some insight into the performance issues of communication networks operating with FACT. To this end, traffic on a VP with FACT will be modeled as a stochastic process at the VC level and at the burst level.

A brief discussion of the general assumptions used for the analysis is presented in the next section. The third section of this chapter presents the models for fixed allocation which will be compared with models for FACT under three traffic configurations. The traffic situation of a single class of flexible calls is considered first at the VC level and the burst level in Sections 6.4 and 6.5, respectively. The traffic model considered in Section 6.6 is a mixture of a class of flexible calls and a class of conventional fixed bandwidth calls. It represents two practical situations: introducing new flexible calls to the same VP with fixed bandwidth calls, or the coexistence of short-haul flexible calls and long-haul flexible calls (i.e., both inside and outside the effective control zone) in the same VP. The case for more than two call classes is investigated in Section 6.7. The last section wraps up this chapter with a brief summary.

6.2 General Assumptions

6.2.1 Measurement Units

Bandwidth can be expressed in many possible units, where such units can be bits per second, ISDN B-Channel (64 Kb/s), STS-3 (155.52 Mb/s) and so on. For simplification of the discussion and for comparison sake, bandwidth, or transmission rate measurement is normalized to one generic ***bandwidth unit*** (BU) in this thesis. The bandwidth allocation will be based on the generic bandwidth unit without the need to specify the actual unit.

The call holding time has been commonly used for many congestion control studies. This study considers the amount of information to be a more appropriate parameter than the holding time for integrated communications. The term call holding time originates from teletraffic studies in telephony. It is a function of transmission rate or bandwidth used and the message length. Many data communication applications now can be accommodated by several different transmission rates, thereby yielding different holding times. For example, it takes more than one minute with a full ISDN primary rate access at 1.536Mb/s to transmit a 100Mb medical image, while it requires less than one second with a full STS-3 at 155Mb/s.

Hence, in this study each communication request is measured by the total amount of information transmitted, as opposed to the call holding time or the transmission time. ***"Call length"*** is defined as the total amount of information of a particular call. The call length could be measured by any information unit such as bit, octet, kilobyte, or megabyte. It is assumed, however, that a very large unit is used so that any finite real number call length is reasonable.

6.2.2 Network Activities

The assumptions for the overall network activities and the network subscribers are the following.

1. The number of subscribers is fairly large compared to the network capacity. This assumption implies that, at the VC level, the population can be considered as infinite.
2. There is no correlation among the subscribers, in other words, each call or burst is independent of the others.
3. A blocked call and burst are rerouted so that they will not return.
4. If traffic overloads occur, they only last for a short period of time. Suppose the mean burst arrival rate for a call is λ, the average burst transmission time at 1 BU is μ^{-1}, and the VP capacity is C; then this assumption requires $\lambda/\mu < C$ for an infinite population, and

$(N - L)\lambda/\mu < C$ for an finite population N, where L is the minimum number of bursts to fully utilize the capacity C.

5. Times for relinquishment and re-allocation are so small compared to the average inter-arrival times and transmission times that they are considered as insignificant or negligible. This assumption is reasonable because of the limited radius of the effective control zone for FACT.

6.2.3 Traffic Model Assumptions

The arrival and departure processes of calls and bursts operates under the following assumptions.

- The call length and burst length are assumed to have exponential distributions.
- Calls and bursts are assumed to arrive according to Poisson processes from single stream and aggregation of finite number of streams, respectively. A burst can be regarded as a call at a much smaller time scale, i.e., a mini-call. The population of calls is the number of subscribers, while the total population of burst traffic streams is the number of all calls in-progress. The call population is assumed to be infinite, because the number of subscribers is very large as compared to the network capacity. The burst population is further assumed to be the average number of calls in-progress. Thereafter, for simplicity, the notion "call" will be used to refer to both burst and call, whenever possible. The actual interpretation is dependent on the level of control. For instance, "call" at the burst level is, in essence, a burst or mini-call, and "call length" is the length of a burst.
- A call can be served with any amount of bandwidth between b_{min} and b_{max} BUs inclusively, where $b_{min} \leq 1 \leq b_{max}$ by the choice of the bandwidth unit. Furthermore, for simplicity of discussion, bandwidth capacity C at the level under study is assumed to be an integer multiple of b_{min} and b_{max}. As stated in the paragraph above, b_{min} represents the minimal bandwidth requirement of flexible calls at the call level, and the minimal bandwidth requirement of a burst at the burst level. The bandwidth units, BUs, are different at these two levels.

The single flexible class traffic model is depicted in Figure 6.15. At the call level, calls arrive according to a Poisson distribution with exponential call length. Each in-progress call is modeled by a Markovian process of two states: idle and busy. A burst arrives at the state transition from idle to busy, and stops at the state transition from busy to idle. It has been assumed that the bursts arrive independently according to Poisson processes from a finite population N which is the average number of in-progress calls. Please note the λ's in the traffic model may have different values for burst traffic and call traffic.

The analytical model is depicted in Figure 6.16 for calls at the call level.

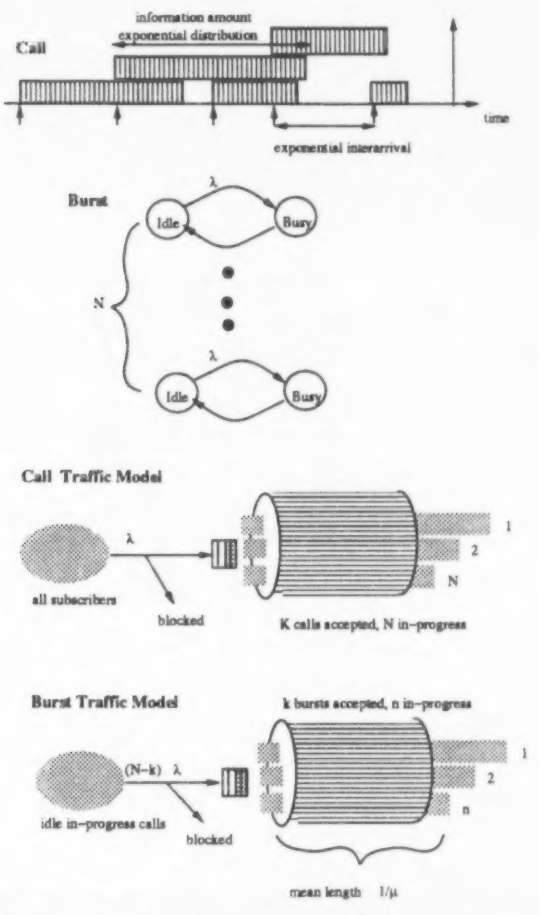

Figure 6.15: Traffic Model for Single Class of Flexible Calls/Bursts

There can be at most $H = C/b_{min}$ calls in progress, and S in the queue, resulting in $H + S$ as the maximal number of calls in the network. A call will be blocked if it arrives after $H + S$ calls are already in the network.

Before the stochastic analytical model can be established, the transmission time distribution has to be investigated for FACT, because it determines the departure rate of a call.

The transmission time is given simply by dividing the call length by the bandwidth if the bandwidth allocation is constant. Because the call length is of exponential distribution with mean $1/\mu$, the transmission time at constant bandwidth b is also exponentially distributed with mean $1/b\mu$. However, if the bandwidth allocation varies during the call period, the distribution of the

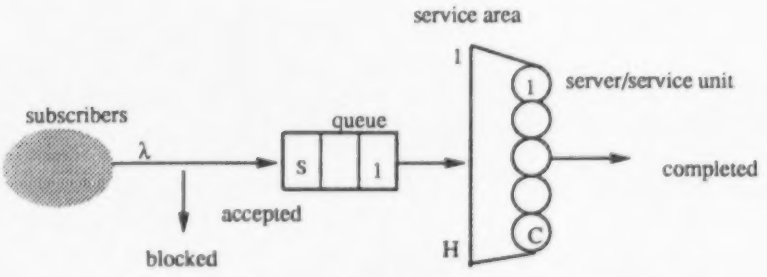

Figure 6.16: Access Control Model for Single Class of Flexible Calls

transmission time warrants more investigation. Lemma 6.1 determines the characteristics of the remaining transmission time after a bandwidth change has occurred at a random instant in the duration of a call.

Lemma 6.1 *The transmission time for the remaining portion of the call, after changing bandwidth to b units at any random instant, is exponentially distributed with mean* $1/b\mu$.

Proof. Let Y be a random variable denoting the remaining amount of information to be transmitted after a random instant t_0 as illustrated in Figure 6.17.

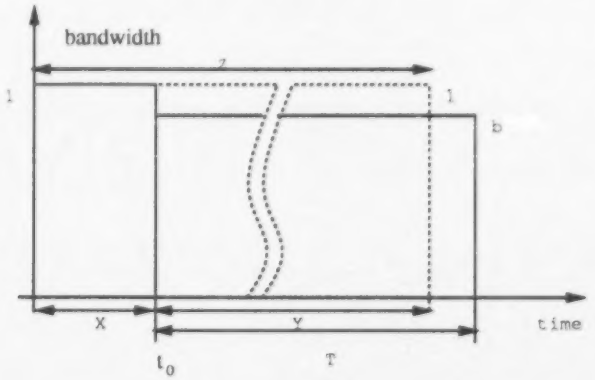

Figure 6.17: Holding Time Distribution

The total information (Z) has an exponential distribution, it is, thus, memoryless [15]. This implies that the remaining information Y, after a random instant t_0, is still exponentially distributed with mean $1/\mu$, with the distribution function being $F(y) = Prob(Y \leq y) = 1 - e^{-\mu y}$, and the associated density function being $f(Y) = \mu e^{-\mu y}$. Let T be a random variable denoting the time

required to transmit Y at the new constant bandwidth b. T is given by $T = Y/b$. Because the bandwidth b is constant, the distribution function for T is then $F(t) = Prob(T \leq t) = Prob(Y/b \leq t) = 1 - e^{-\mu b t}$, and the density function $f(t) = \mu b e^{-\mu b t}$. The remaining transmission time T is, therefore, also exponentially distributed with mean $1/(b\mu)$.

Applying this process recursively, it can be shown that the remaining transmission time after the bandwidth has been changed at another instant t_1 from b to b' is still exponentially distributed with mean $1/(\mu b')$. That concludes the proof. □

There is no restriction in the proof on the value of b, although only a case for $b < 1$ is depicted in Figure 6.17.

Lemma 6.1 is a result of the exponential call length assumption only. It is applicable regardless of the relinquishment policies or call classes. This lemma will serve as the analytical basis of the traffic model.

6.3 Comparable Fixed Allocation Models

FACT will be compared to the conventional schemes using the same traffic and network capacity, C. Although a CPE may be capable of supporting multiple transmission rates, usually a fixed bandwidth is allocated with the existing schemes, and used for the whole duration of a call. An access control policy determines whether or not to accept a call request and how to allocate bandwidth to an accepted call. Existing policies all allocate fixed amounts of bandwidth to a call for its duration. The exact allocation varies according to the control objectives, thereby resulting in different policies. With the new flexible scheme, the allocation can be at the maximum, the minimum or any amount in between. In this thesis, three policies are considered as representatives of conventional policies for comparison with FACT, where the CPE can support any rate between b_{min} and b_{max} BUs.

The first policy is intended to minimize the blocking probability by assigning to each communication request the minimum bandwidth, so that the maximal number of calls/bursts can be served with the given total capacity, C BUs, at the cost of probably increased transmission delay. This policy is more in the interest of the network operators than the users of in-progress calls. This policy is referred to as *Minimum Allocation* and is denoted by *Min* in the presentation of numerical results. Its analytical models are M/M/H/L+S/N queues, with $N = \infty$ at the VC level, $N < \infty$ at the burst level.

The next policy deals with the other extreme. It permits each call/burst to use maximum bandwidth in an attempt to minimize the transmission delay, at the cost of probably increased blocking rate and decreased bandwidth utilization. This policy is more attractive to the users of in-progress calls than to the network operators. It is referred to as *Maximum Allocation* with notation

Max. The corresponding analytical model are M/M/L/L+S/N queues.

The last policy lies between these two extreme policies. It allocates one BU to each call to balance transmission delay and blocking rate, or in fact, users' and network operator's interest. This policy is referred to as *Average Allocation* with notation *Avg*. The corresponding analytical models are M/M/C/C+S/N queues.

The steady state probabilities of these processes can be obtained from many textbooks on basic queueing theory, e.g., [39]. For example, the steady probabilities ($p_{k,max}$) and call blocking probability ($P_{bk,max}$) for the Maximum Allocation policy at the VC level are:

$$p_{k,max} = \begin{cases} p_{0,max}\rho_{max}^k/k! & for\ 0 \le k \le L \\ p_{0,max}\rho_{max}^k/(L!L^{L-k}) & for\ L < k \le L+S \end{cases} \tag{6.1}$$

$$p_{0,max}^{-1} = E_L(\rho_{max}) + \sum_{k=1}^{S} \rho_{max}^{L+k}/(L!L^k) \tag{6.2}$$

$$P_{bk,max} = p_{0,max}\rho^{L+S}/(L!b_{max}^{L+S}L^S) \tag{6.3}$$

where $\rho_{max} = \rho/b_{max} = \lambda/(\mu b_{max})$ is the offered load from a single traffic stream to the link with L circuits of capacity b_{max} each, and $E_n(x) = \sum_{i=0}^{n} x^i/i!$ is the incomplete exponential function. The probability $p_{0,max}$ serves as a normalization factor so that the sum of all steady state probabilities is 1. At the burst level, the steady state probabilities and the burst blocking probability for the Maximum Allocation policy are

$$p_{k,max} = \begin{cases} p_{0,max}\binom{N}{k}\rho_{max}^k & for\ 0 \le k \le L \\ p_{0,max}\binom{N}{k}\rho_{max}^k \frac{k!}{L!L^{k-L}} & for\ L < k \le L+S \end{cases} \tag{6.4}$$

$$p_{0,max}^{-1} = \sum_{k=0}^{L}\binom{N}{k}\rho_{max}^k + \sum_{k=L+1}^{L+S}\binom{N}{k}\rho_{max}^k \frac{k!}{L!L^{k-L}} \tag{6.5}$$

$$P_{bk,max} = p_{0,max}\binom{N}{L+S}\rho_{max}^k \frac{(L+S)!}{L!L^S} \tag{6.6}$$

The Average and Minimum Allocations have similar formulas with L replaced by C and H, ρ_{max} by ρ and ρ_{min}, respectively.

6.4 Single Flexible Class at the VC Level

This section considers that communication requests from all subscribers have the same stochastic characteristics. In other words, they are from the same class.

FACT is designed to achieve better bandwidth utilization and blocking rate than the fixed allocation schemes. These intended improvements are proved analytically in this section in the form

of two theorems. The new activity introduced by the FACT scheme is bandwidth relinquishment. One important question is that of the frequency of relinquishments. This is answered by the study of relinquishment probabilities. One possible drawback for delayed relinquishment policies could be longer call delays caused by frequent queueing, as compared to the fixed allocation schemes. Performance measures are, hence, studied for the delayed relinquishment policy, and the results will clear up this doubt. Some numerical examples are provided to illustrate the analytical results. Practical applications of these results are illustrated in Chapter 9.

6.4.1 Assumptions and Model

The basic assumptions and conditions used in this section have been described in the previous section. The relinquishment policies considered will be immediate relinquishment and delayed relinquishment as defined in Chapter 3. They are summarized here for the single class of flexible calls in three categories: arrival and departure processes, call arrival handling and call departure handling.

Call Arrival and Departure Processes

- Calls are assumed to arrive according to a Poisson process from an infinite population, where call lengths have exponential distribution.
- A call can be served with any amount of bandwidth between b_{min} and b_{max} BUs inclusively, where $b_{min} \leq 1 \leq b_{max}$. It is further assumed for simplicity of discussion that the VP bandwidth capacity C is an integer multiple of b_{min} and b_{max}.

Call Arrival Handling

- A call will be accepted immediately upon its arrival, if there are at least b_{max} BUs available.
- If all the bandwidth is used up by the existing calls, an arriving call will be queued in a holding queue of length S until this queue is full.
- If the holding queue is full and each in-progress call is using more than b_{min} bandwidth units, then the in-progress calls will relinquish necessary bandwidth for the new call to be accepted. If the number of in-progress calls is $N_p \leq C/b_{min} - S - 1$, the arriving call will be admitted together with all calls in the queue as a batch; otherwise, the first $C/b_{min} - N_p$ calls in the queue will be admitted and the rest of the calls remain in the queue with the arriving call at the tail of the queue.
- If the queue is full and each in-progress call is using b_{min} bandwidth units, then the arriving call is blocked and never returns.

- Even relinquishment policy is used; that is, each in-progress call gives up the same amount of bandwidth so that each call, including the new one, will occupy the same amount of bandwidth after the new call is accepted.
- Relinquishment is instantaneous.

Call Departure Handling

- As soon as a call departs, the call at the head of the queue will be admitted, if the queue is not empty.
- If the queue is empty at the departure of a call, the bandwidth allocation for each call will be increased until each call uses b_{max} units.
- Bandwidth increase is instantaneous.

As defined in Chapter 4, ***immediate relinquishment policy*** refers to the policy without a holding queue ($S = 0$); in other words, relinquishment is required as soon as an arriving call finds that all bandwidth is used up. A system with $S \geq 1$ is referred to as having a ***delayed relinquishment*** policy.

The notations used in this section are summarized in Table 6.6.

Let the state k be the number of calls in the system. The only possible state transitions from k are to $k+1$ and to $k-1$ because of the assumptions associated with the arrival, departure and instantaneous relinquishment and bandwidth re-allocation. The Markovian property holds for this stochastic process, because both arrivals and departures are exponential processes, and are therefore memoryless [15]. The model is, thus, a one-dimensional birth-death process. Its state diagram is shown in Figure 6.18 together with those of fixed allocation policies. Because of the infinite population assumption, the arrival rate for each state is λ. The departure rate for each state is given in Lemma 6.2.

Lemma 6.2 *The departure rate from state k, μ_k, of the modeled process for FACT is given by*

$$\mu_k = \begin{cases} k\mu_{max} & \text{if } 1 \leq k \leq L \\ C\mu & \text{if } L < k \leq H + S \end{cases}$$

for both the immediate and delayed relinquishment policies.

Proof. The first expression is true for any relinquishment policy. Because for $k \leq L$, each of the k calls uses its maximal allowed (b_{max}) bandwidth units, resulting in the individual departure rate $\mu_{max} = \mu b_{max}$ according to Lemma 6.1. Thus, for the state $k \leq L$, the departure rate is $k\mu_{max}$.

Table 6.6: Summary of Notations for Single Class Call Study

C	VP bandwidth capacity,
L	$= C/b_{max}$ the minimal number of calls to fully utilize the VP,
H	$= C/b_{min}$ the maximal number of calls to fully utilize the VP,
N_p	number of calls in progress
N_q	number of calls in the queue
k	$= N_p + N_q$, state of the birth-death process, number of calls in the network
$\lambda(k)$	mean arrival rate when the state is k,
$1/\mu$	mean amount of information,
μ_{min}	$= \mu b_{min}$, call departure rate when b_{min}, instead of 1, BUs are used
μ_{max}	$= \mu b_{max}$, call departure rate when b_{max}, instead of 1, BUs are used,
μ_k	state transition rate from k to $k-1$,
ρ	$= \lambda/\mu$,
ρ_{min}	$= \lambda/\mu_{min}$,
ρ_{max}	$= \lambda/\mu_{max}$,
u	$= \rho/C = \rho_{max}/L = \rho_{min}/H$,
u_{min}	$= \rho_{min}/C$,
u_{max}	$= \rho_{max}/C$,
S	queueing threshold before relinquishment
h	$= H + S - L$ number of states from L to $H + S$
D_p	call holding time, or transmission time
D_q	call queueing time
D	$= D_p + D_q$, call delay, the time spent in the network
p_k	equilibrium probability of system being in state k
P_k	$= \sum_{i=0}^{k} p_i$, the equilibrium probability of that system being in state $j \leq k$

Let N_p denote the number of in-progress calls. For $k > L$, each of the N_p in-progress calls uses C/N_p units of bandwidth. Therefore each has an individual departure rate of $\mu C/N_p$ according to Lemma 6.1. The departure rate for state $k > L$, then, is $N_p \cdot \mu C/N_p = C\mu$. □

As a result of Lemma 6.2, we can obtain the steady state probabilities independent of the relinquishment policy.

Lemma 6.3 *The steady state probability, p_k, for the system being in state k is given by*

$$p_k = \begin{cases} \left[E_L(\rho_{max}) + u\frac{\rho_{max}^L}{L!}\frac{1-u^{H+S-L}}{1-u}\right]^{-1} & for\ k = 0 \\ p_0\rho_{max}^k/k! & for\ 0 < k \leq L \\ p_L u^{k-L} & for\ L < k \leq H+S \end{cases}$$

where $E_n(x) = \sum_{i=0}^{n} x^i/i!$ is the incomplete exponential function.

Proof. Applying Lemma 6.2 to the general birth-death process state probabilities $p_k = p_0 \prod_{i=0}^{k-1} \lambda_i/\mu_{i+1}$, we obtain, for $0 < k \leq L$

$$p_k = p_0 \prod_{i=0}^{k-1} \lambda/\mu_i$$

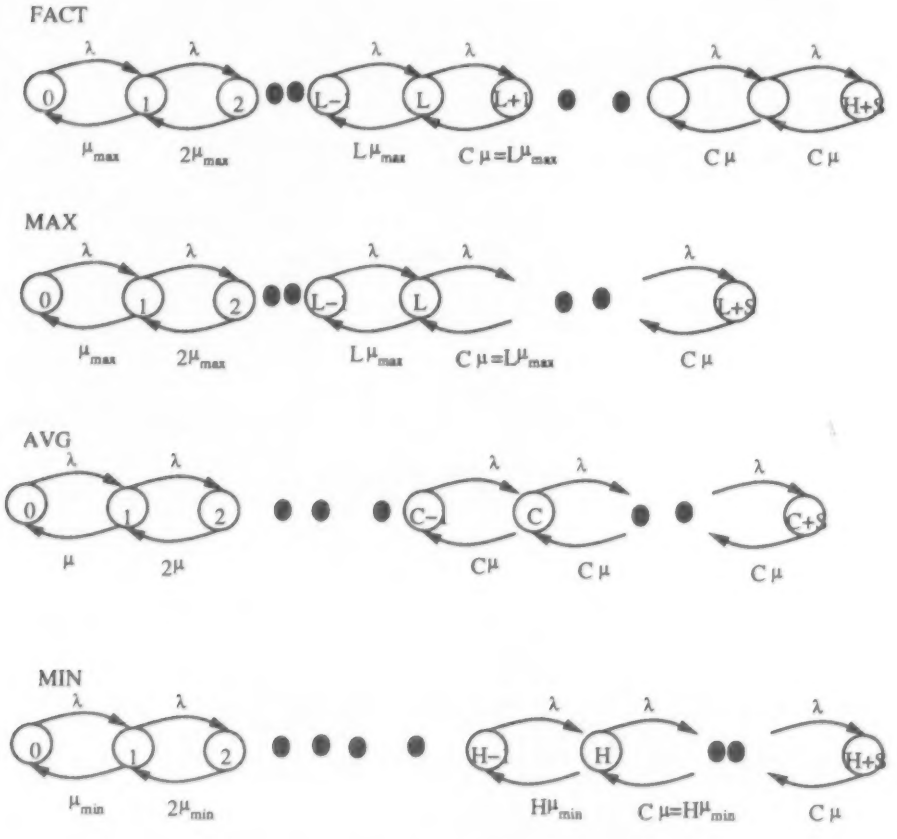

Figure 6.18: State Diagram for Single Class of Flexible Calls

$$= p_0 \prod_{i=0}^{k-1} \lambda/((i+1)\mu_{max})$$
$$= p_0 \rho_{max}^k / k!$$

and for $L < k \leq H + S$,

$$p_k = p_0 \prod_{i=0}^{L-1} \frac{\lambda}{(i+1)\mu_{max}} \prod_{i=L}^{k-1} \frac{\lambda}{C\mu}$$
$$= p_L u^{k-L}$$

Since $\sum_{k=0}^{H+S} p_k = 1$,

$$p_0^{-1} = E_L(\rho_{max}) + \frac{p_L}{p_0} \sum_{i=L+1}^{H+S} u^{i-L}$$

$$
\begin{aligned}
&= E_L(\rho_{max}) + \frac{\rho_{max}^L}{L!} \sum_{i=1}^{H+S-L} u^i \\
&= E_L(\rho_{max}) + \frac{\rho_{max}^L}{L!} u \sum_{i=0}^{H+S-L-1} u^i \\
&= E_L(\rho_{max}) + u\frac{\rho_{max}^L}{L!} \frac{1-u^{H+S-L}}{1-u}
\end{aligned}
$$

□

Figure 6.19 depicts the steady state probability distributions for FACT and three comparable fixed allocation policies with $u = 0.9, C = 100, b_{min} = 0.75, b_{max} = 1.25$ and $S = 0$. The maximal number of calls which can be served by each policy is $H = 133$ for FACT and the Minimum Allocation, $L = 80$ for the Maximum Allocation, and $C = 100$ for the Average Allocation.

The figure shows that all the fixed allocation policies have relatively high probabilities in the region close to their respective maximal numbers of calls which can be served, whereas FACT has high probabilities close to the center of the state spectrum. In other words, there are high likelihoods for the fixed allocation policies to have number of in-progress calls close the limit, and for FACT to have relatively more free capacity to take on more calls. The steady state probabilities, hence, suggest that FACT would have lower blocking and higher utilization than the conventional fixed policies. The remainder of this section gives an analytical proof for this observation, and also studies other interesting performance issues such as call delay, queueing performance, and the additional (relinquishment and re-allocation) cost for FACT.

Once the steady state probabilities have been obtained, the next step is to derive expressions for various performance measures for FACT and to compare them to existing schemes.

6.4.2 Performance Measures and Comparisons

A. Call Blocking Rate

Denote $h = H + S - L$ as the number of states between L and the last state $H + S$.

Lemma 6.4 *Let P_{bk} denote the probability that a flexible call from a single class with offered traffic density ρ is blocked at a VP with capacity C. Then P_{bk} is given by*

$$P_{bk} = \frac{\rho_{max}^L}{L!} u^h p_0 \tag{6.7}$$

And, when $S = 0$ the call blocking probability becomes

$$P_{bk}|_{S=0} = \frac{u^H L^L}{L!} p_0 \tag{6.8}$$

where $u = \rho/C$.

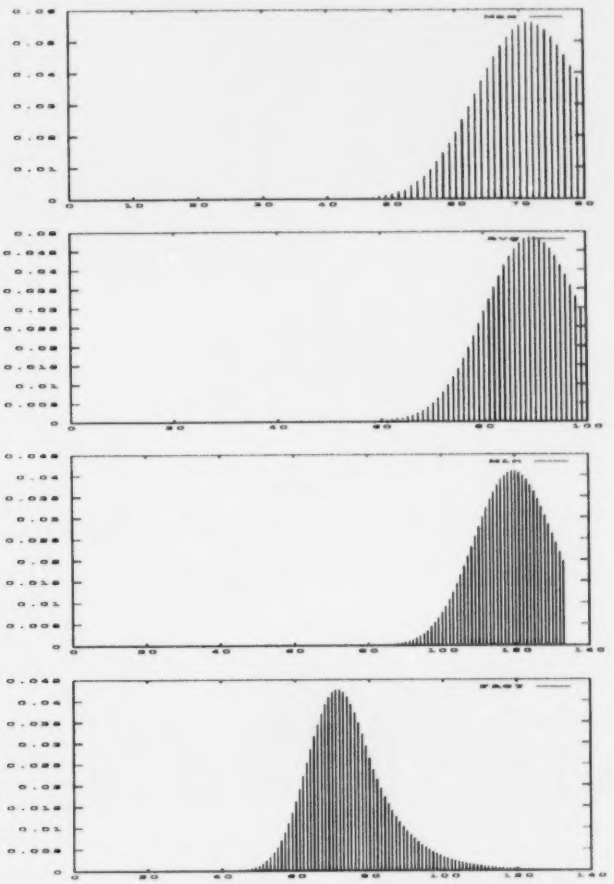

Figure 6.19: Probability Distributions Versus Call Numbers for Single Class of Flexible Calls

Proof. A call request is blocked when, and only when, there are already $H + S$ calls in the network; that is, H calls in progress, and S calls waiting in the queue. Therefore, the probability for an arriving call being blocked is

$$\begin{aligned} P_{bk} &= p_{H+S} \\ &= p_L u^h \\ &= p_0 \frac{\rho^L}{L! b_{max}^L} u^h \\ &= p_0 \frac{r_{max}^L}{L!} u^h \end{aligned}$$

Since $b_{max}L = C$ and $u = \rho/C$ by definition, the blocking rate with immediate relinquishment policy then is

$$\begin{aligned} P_{bk}|_{S=0} &= \frac{\rho^{L+h}}{L!b_{max}^{L}C^{h}}p_0 \\ &= \frac{\rho^{H}}{L!b_{max}^{L}C^{H-L}}p_0 \\ &= \frac{\rho^{H}C^{L}}{L!b_{max}^{L}C^{H}}p_0 \\ &= \frac{u^{H}L^{L}}{L!}p_0 \end{aligned}$$

□

The next theorem proves the advantage of FACT over fixed allocations. The proof was obtained with the help of Dr. B. Schmuland.

Theorem 6.1 *FACT always has lower blocking probability than any fixed allocation policies.*

Proof. The proof is obtained in two steps. The first step establishes that the Minimum Allocation has the lowest blocking probability among all fixed allocation policies. Then, the second step further shows that FACT has an even lower blocking probability than the Minimum Allocation policy.

Any fixed allocation policy can be viewed as M/M/W/W+S models, where W is the maximal number of calls that can be served by that allocation policy with fixed capacity C bandwidth units.

The average arrival rate λ, mean call length $1/\mu$, bandwidth capacity C and queue length S are given as fixed values for all policies. Different bandwidth allocations result in different departure rates, $\mu_w = C\mu/W$, and $\rho_w = \lambda/\mu_w$. The bandwidth allocated to each call is fixed at μ_w/μ bandwidth units for the duration of the call. Regardless of the policy W, however, $C\mu = W\mu_w$ and the utilization factor $u = \rho_w/W = \rho/C$ are still fixed.

The goal of the first step of the proof is to find which fixed allocation policy has the lowest blocking probability $P_{bk}(W)$ for the generic policy model M/M/W/W+S, i.e., to find the W such that $P_{bk}(W)$ is the smallest.

The blocking probability for any policy is in fact the probability of the system being at the last state $W + S$.

$$\begin{aligned} P_{bk}(W) &= \frac{\rho_w^{W+S}/(W!W^S)}{\sum_{k=0}^{W}\rho_w^k/k! + \rho_w^W/W!\sum_{k=1}^{S}u^k} \\ &= \frac{\rho_w^S/W^S}{(W!/\rho_w^W)\sum_{k=0}^{W}\rho_w^k/k! + \sum_{k=1}^{S}u^k} \end{aligned}$$

The only term not fixed in $P_{bk}(W)$ is

$$\begin{aligned}
& \frac{W!}{\rho_w^W} \sum_{k=0}^{W} \rho_w^k / k! \\
= & \frac{W!}{(Wu)^W} \sum_{k=0}^{W} (Wu)^k / k! \\
= & \sum_{k=0}^{W} \frac{(W-k)!(Wu)^k}{(Wu)^W} \\
= & \sum_{k=0}^{W} \frac{\overbrace{W(W-1)\cdots(W-k+1)}^{k}}{W^k} u^{k-W}
\end{aligned}$$

It is increasing in W, as each term of the summation is increasing in W. Therefore, the blocking probability, $P_{bk}(W)$, decreases as W increases (see Figure 6.20). In other words, the fixed allocation serving the most number of calls, or allocating the minimal number of bandwidth to each call, has the lowest blocking probability. That policy is the Minimum Allocation policy.

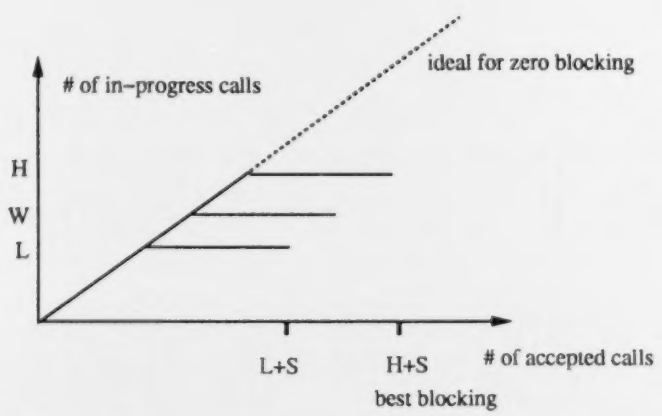

Figure 6.20: Lowest Blocking Among All Fixed Allocation Policies

The second step is then to find the best policy among all those policies having the maximal number of states, $H + S$. This group of policies is modeled by M/M/W/H+S with fixed arrival rate λ, a call departure rate μ_w as a function of W, and W now being the minimal number of calls to fully utilize the capacity. Again, $W\mu_w = C\mu$ is fixed. So is $u = \rho_w/W = \rho/C$. For FACT the minimal number of calls that can be served by capacity C is $W = L$ and $\mu_w = \mu_{max}$. For the Minimum Allocation, it is $W = H$ and $\mu_w = \mu_{min}$. The blocking probability for such a system is

$$P_{bk}(W) = \frac{\frac{\rho_w^W}{W!} u^{H+S-W}}{\sum_{k=0}^{W} \rho_w^k/k! + (\rho_w^W/W!) \sum_{k=1}^{H+S-W} u^k}$$

$$
\begin{aligned}
&= \frac{\frac{W^W}{W!}u^{H+S}}{\sum_{k=0}^{W}\rho_w^k/k! + (\rho_w^W/W!)\sum_{k=1}^{H+S-W}u^k} \\
&= \frac{u^{H+S}}{\frac{W!}{W^W}\sum_{k=0}^{W}\rho_w^k/k! + u^W\sum_{k=1}^{H+S-W}u^k} \\
&= \frac{u^{H+S}}{\sum_{k=0}^{W}\frac{W!}{W^W}\rho_w^k/k! + \sum_{k=W+1}^{H+S}u^k} \\
&= \frac{u^{H+S}}{\sum_{k=0}^{W}\frac{W!W^k}{k!W^W}u^k + \sum_{k=W+1}^{H+S}u^k}
\end{aligned}
$$

The denominator always contains $H+S$ terms regardless of W. But the coefficient of u^k in the first summation for $k \leq W$

$$\frac{W!W^k}{W^W k!} = \frac{\overbrace{W(W-1)\cdots(k+1)}^{W-k}}{W^{W-k}}$$

increases as W decreases. Therefore, $P_{bk}(W)$ decreases as W decreases (see Figure 6.21 where W at the circle is the bend). The minimum blocking is achieved at the minimal W, i.e., with FACT.

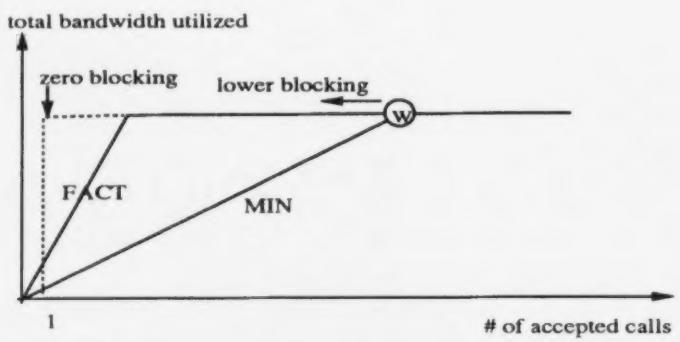

Figure 6.21: Lowest Blocking Policy Among Those Serving Maximal Number of Calls

This concludes the proof □

The first step of the proof actually indicates that, given capacity C and traffic characteristic λ and μ, it is best to be able to serve as many calls as possible. Referring back to Figure 6.18, it means that the fixed allocation policy with the lowest blocking is the one with the largest number of states. In terms of bandwidth allocation, that condition translates into one in which if we could give each call an infinitely small amount of bandwidth, then blocking probability would approach zero. Given the condition that a call can be served by b_{min} at minimum, the largest number of calls that can be served by C BUs is $H+S$. Both FACT and the Minimum Allocation satisfy it.

The second step shows that blocking probability can be lowered by moving the bend of the utilization curve to the left in Figure 6.21. It means that the best policy among those policies which

is able to serve the largest number of calls, $H+S$, is to fully utilize the bandwidth with the minimal number of calls. The lowest blocking policy is the one with state diagram reaching the $C\mu$ departure rate the earliest. Ideally the best policy would be $W = 1$, that is, to assign all bandwidth to a single call, and thus "flushing" it out in the shortest time. Given the practical constraint that a call can accept b_{max} BUs at the maximum, instead of all C BUs, the best policy in reality is therefore the policy with $W = L = C/b_{max}$.

Hence, for a policy to have the lowest blocking probability among *all* policies, fixed or otherwise, it must satisfy both these criteria, i.e., it must have the largest number of states, and its utilization curve must reach 1 with the minimal number of calls. Therefore, it can be further said that FACT has the lowest blocking probability among *all* bandwidth allocation policies.

The next theorem further quantifies the improvement that FACT offers in terms of the blocking probability over the fixed allocation policies.

Theorem 6.2 *When $b_{min} < b_{max}$, the ratios of blocking rate of FACT to that of the Maximal and Average Allocation policies have upper bounds u^{H-L} and u^{H-C}, respectively. In other words, the blocking rate of FACT is less than $u^{H-L} < 1$ or $u^{H-C} < 1$ times that of these two fixed allocation policies, respectively.*

Proof. Let B_{max}, B_{avg} denote the ratios of the blocking probability of FACT to that with Maximum and Average Allocation policies, respectively.

First, the proof for the comparison with Maximum Allocation is the following. By definition $b_{max}L = C$ and $\rho/C = u$, therefore

$$\begin{aligned} B_{max} &= \frac{\frac{\rho^{L+h}}{L!b_{max}^{L}C^{h}}p_0}{\frac{\rho^{L+S}}{L!b_{max}^{L+S}L^{S}}p_{0,max}} \\ &= u^{h-S}\frac{p_0}{p_{0,max}} \\ &= u^{H-L}\frac{p_{0,max}^{-1}}{p_0^{-1}} \end{aligned}$$

As stated early in Equation (6.2),

$$\begin{aligned} p_{0,max}^{-1} &= E_L(\rho_{max}) + \sum_{k=L+1}^{L+S}\frac{\rho^k}{L!b_{max}^{L}C^{k-L}} \\ &= E_L(\rho_{max}) + \sum_{k=L+1}^{L+S}\frac{u^{k-L}\rho^L}{L!b_{max}^{L}} \end{aligned}$$

Lemma 6.3 implies that

$$p_0^{-1} = E_L(\rho_{max}) + \sum_{k=L+1}^{H+S}\frac{u^{k-L}\rho^L}{L!b_{max}^{L}}$$

$$> p_{0,max}$$

Since $u < 1$ by the temporary overload assumption,

$$B_{max} < u^{H-L} < 1$$

A similar result is obtained for the comparison with the Average Allocation policy. Since $b_{max} > 1$, $b_{max}^L > b_{max}^k$ for $k = 0, ..., L-1$, the ratio of the FACT blocking probability to that of Average Allocation is then

$$\begin{aligned} B_{avg} &= \frac{\frac{\rho^{H+S}}{L!b_{max}^L C^{k-L}} p_0}{\frac{\rho^{C+S}}{C!C^{S}} p_{0,avg}} \\ &= \frac{C!\rho^{H-C} p_{0,avg}^{-1}}{L!C^{H-L} b_{max}^L p_0^{-1}} \\ &= u^{H-C} \frac{p_{0,avg}^{-1}}{\frac{L!C^{C-L}}{C!} b_{max}^L p_0^{-1}}. \end{aligned}$$

Since

$$\begin{aligned} b_{max}^L p_0^{-1} &= b_{max}^L E_L(\rho_{max}) + b_{max}^L \sum_{k=L+1}^{H+S} \frac{\rho^k}{L!b_{max}^L C^{k-L}} \\ &> E_L(\rho) + b_{max}^L \sum_{k=L+1}^{H+S} \frac{\rho^k}{L!b_{max}^L C^{k-L}} \\ &= E_L(\rho) + \sum_{k=L+1}^{H+S} \frac{\rho^k}{L!C^{k-L}} \\ &> E_L(\rho) + \sum_{k=L+1}^{C} \frac{\rho^k}{L!C^{k-L}} + \sum_{k=C}^{C+S} \frac{u^{k-C}\rho^C}{C!}. \end{aligned}$$

Noting that $\frac{L!/C^L}{C!/C^C} \geq 1$, and $\frac{C^C}{C!} > \frac{C^k}{k!}$ for $k < C$,

$$\begin{aligned} \frac{L!C^{C-L}}{C!} b_{max}^L p_0^{-1} &> E_L(\rho) + \frac{L!C^{C-L}}{C!} \sum_{k=L+1}^{C} \frac{\rho^k}{L!C^{k-L}} + \sum_{k=C}^{C+S} \frac{u^{k-C}\rho^C}{C!} \\ &= E_L(\rho) + \sum_{k=L+1}^{C} \frac{\rho^k C^C}{C!C^k} + \sum_{k=C}^{C+S} \frac{u^{k-C}\rho^C}{C!} \\ &> \sum_{k=0}^{C} \frac{\rho^k}{k!} + \sum_{k=C}^{C+S} \frac{u^{k-C}\rho^C}{C!} \\ &= p_{0,avg}^{-1} \end{aligned}$$

Therefore,

$$B_{avg} < u^{H-C}$$

B_{avg} is further bounded by 1 because $u < 1$. □

The improvement is at least u^{H-L} and u^{H-C} times, respectively. These two bounds are u^{H-W} with W being the maximal number of calls that can be accommodated by that fixed allocation policy. The improvement is, thus, exponential in $C(1/b_{min} - 1/b_{max})$. This seems to agree with the observation made in the previous chapter for static allocation that FACT is more effective for broadband networks or traffic aggregated from a large number of narrow-band calls.

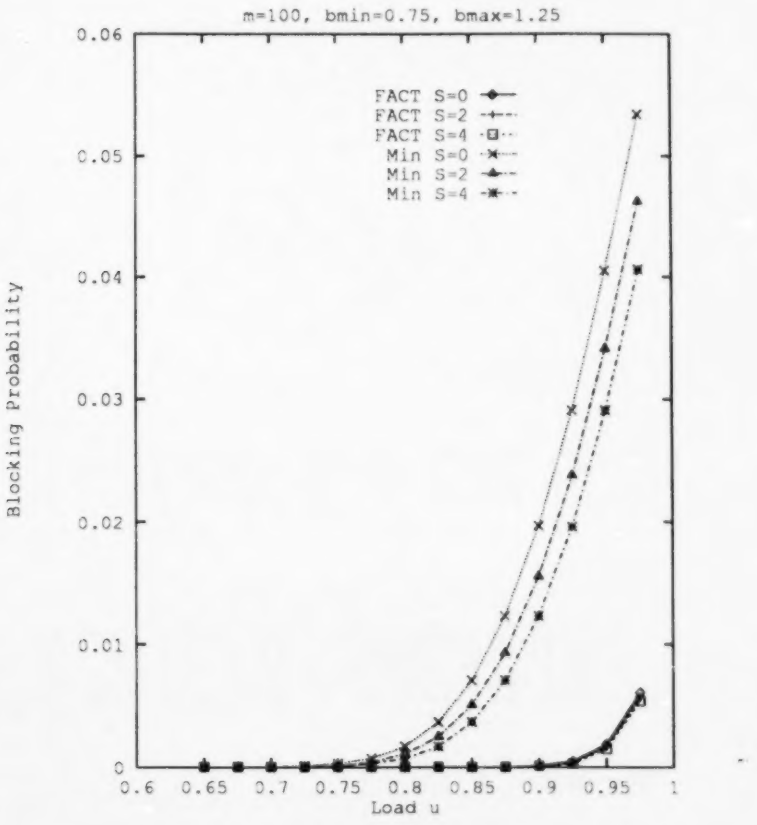

Figure 6.22: Call Blocking Probability Comparison: FACT vs. Minimum Allocation

Due to the arbitrary selection of the unit of BU, another significance of this theorem is that the improvement bound can be applied to any fixed allocation. In other words, for any fixed allocation at bandwidth C/W, the blocking probability improvement for FACT is at least u^{H-W} times.

Figures 6.22 shows the call blocking probability comparisons between FACT and the Minimum Allocation policy with $b_{min} = 0.75, b_{max} = 1.25$. In this figure, the blocking probabilities for FACT

with different queue lengths are too close to be plotted into distinctive lines. This seems to suggest that, while queueing for the Minimum Allocation does improve the blocking rate, it, however, has little effect with FACT.

Observation 6.1 *The most effective way to reduce the blocking rate then is to increase b_{max}, not to increase the queue length S, given the traffic load and VP capacity.*

B. Expected Bandwidth Utilization

Bandwidth utilization is defined as the ratio of occupied bandwidth to total bandwidth for a stationary system, and is denoted by the random variable U.

Intuitively one would expect the bandwidth utilization for FACT to have exactly the same form, $E(U) = u(1 - P_{bk})$, as that for the fixed allocations. Since $\lambda(1 - P_{bk})$ is the average *carried traffic*, then $\rho(1 - P_{bk})$ is the average number of occupied bandwidth units. The latter is divided by the total bandwidth, C, to yield the average bandwidth utilization. The only difference among the policies comes from their probability distributions; or more specifically, the blocking probability P_{bk}.

The bandwidth utilization $E(U) = u(1 - P_{bk})$ for FACT is verified analytically below.

The number of occupied bandwidth units is kb_{max} for state $k \leq L$, and C for state $L < k \leq H + S$. Thus,

$$E(U) = (\sum_{k=0}^{L} b_{max} k p_k + \sum_{k=L+1}^{H+S} C p_k)/C$$

By definition, $b_{max}\rho_{max} = \rho$. From Lemma 6.3, and $kp_k = \rho_{max} p_{k-1}$ for $k \leq L$, $Cp_k = \rho p_{k-1}$ for $L < k \leq H + S$, therefore, the expected utilization can be obtained as

$$\begin{aligned} E(U) &= (\sum_{k=0}^{L} b_{max} k p_k + \sum_{k=L+1}^{H+S} C p_k)/C \\ &= (b_{max}\rho_{max} \sum_{k=0}^{L} p_{k-1} + \rho \sum_{k=L+1}^{H+S} p_{k-1})/C \\ &= (\rho \sum_{k=0}^{L} p_{k-1} + \rho \sum_{k=L+1}^{H+S} p_{k-1})/C \\ &= u \sum_{k=0}^{H+S-1} p_k \\ &= u(1 - P_{bk}) \end{aligned}$$

with p_{-1} defined as 0.

Applying this result to Theorem 6.1, it can be shown immediately that FACT outperforms the fixed allocation policies in bandwidth utilization as well.

Theorem 6.3 *FACT always result in higher expected bandwidth utilization than any fixed allocation policies.*

So far, it has been proved that FACT outperforms conventional fixed allocation policies in both blocking rate and bandwidth utilization. The next two subsections deal with comparisons for queueing performance and for call delays, respectively.

C. Queue Length and Queueing Delay

A holding queue is used for delayed relinquishment policies. This queue introduces extra queueing delay to call delay, and also requires provision of storage. The following problems then need to be studied here.

- queueing probability – the likelihood that an arriving call has to be queued,
- queue occupancy – how many calls are waiting before the arriving call,
- average queueing delay – how much time on average a call has to spend in queue

One of three possible actions can be taken for an arriving call: to block the call, to admit the call without queueing, and to hold the call in the queue.

The probability of the first event, i.e., blocking, has already been given as P_{bk}. To find the probability for a call being queued upon its arrival amounts to finding the probability of admission without waiting in the queue.

An arriving call is admitted without queueing if there are either $k < L$ calls in progress, or $k > L$ calls in progress but this arriving call triggers a relinquishment, and, therefore, is admitted without queueing. The probability for the first event ($k < L$) is P_L. The second event, direct admission after triggering relinquishment, occurs when the queue is full and $k < H$. Its probability, thus, is

$$\sum_{i=1}^{s} p_{L+i(S+1)-1} = \frac{1-u^{s(S+1)}}{1-u^{S+1}} p_{L+S}$$

where $s = \frac{h}{S+1}$ is the minimal number of batch admissions required to reach state H from the idle state L. For simplicity of discussion, it is assumed in this thesis that h and S are chosen such that h is an integer multiple of $S+1$.

Therefore, the probability for an arriving call to enter the network without waiting is given by

$$P(no\ waiting) = P_{L-1} + \frac{1-u^{s(S+1)}}{1-u^{S+1}} p_{L+S} \tag{6.9}$$

An arriving call can either be accepted without queueing, accepted with queueing, or blocked. Therefore, the probability for an arriving call being queued is given by

$$\begin{aligned} P(queued) &= 1 - P(no\ waiting) - P_{bk} \\ &= 1 - P_{L-1} - \frac{1-u^{s(S+1)}}{1-u^{S+1}} p_{L+S} - p_L u^h \end{aligned}$$

When $S = 0$, $P(queued) = 0$ as it should.

Figures 6.23, compares the queueing probabilities between FACT and the Average Allocation policy with $S = 1, 3, 5$.

Given these quite large queueing probabilities, the queueing delay and the total call delay must be investigated in order to determine the usefulness of FACT. FACT would not be attractive to many applications, should it introduce longer call delays than conventional policies; even though it does improve call blocking rate and bandwidth utilization.

The queue length, N_q, distribution can be derived from the steady state probabilities of the whole system. For integer $1 \le j \le S$

$$\begin{aligned} Prob(N_q = j) &= \sum_{(k-L) mod (S+1) = j} p_k \\ &= \sum_{i=0}^{s-1} p_{L+i(S+1)+j} \\ &= p_L u^j \sum_{i=0}^{s-1} u^{i(S+1)} \\ &= p_L u^j \frac{1-u^{s(S+1)}}{1-u^{S+1}} \end{aligned}$$

Since $s = h/(S+1)$ by definition, for $1 \le j \le S$,

$$Prob(N_q = j > 1) = p_L \frac{1-u^h}{1-u^{S+1}} u^j \tag{6.10}$$

$$Prob(N_q = 0) = 1 - p_L \frac{(1-u^h)u(1-u^S)}{(1-u^{S+1})(1-u)} \tag{6.11}$$

Average queue length can be obtained

$$\begin{aligned} E(N_q) &= \sum_{j=1}^{S} j Prob(N_q = j) \\ &= p_L \frac{1-u^h}{1-u^{S+1}} \sum_{j=1}^{S} j u^j \end{aligned}$$

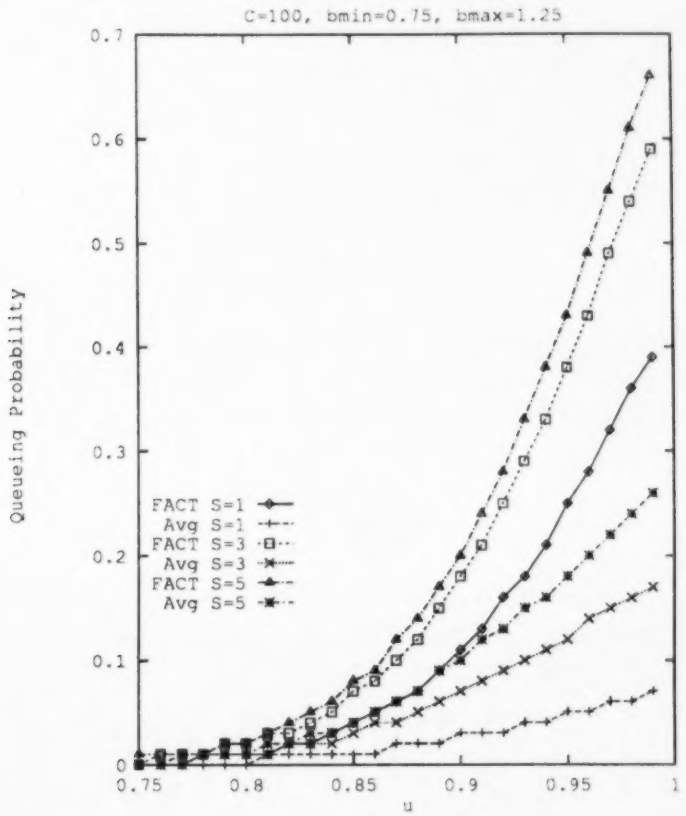

Figure 6.23: Queueing Probabilities For Single Flexible Call Class

Therefore, the expected number of calls in the queue is

$$E(N_q) = p_L \frac{1-u^h}{1-u^{S+1}} \frac{1-Su^{S-1}+(S-1)u^S}{(1-u)^2} \tag{6.12}$$

The average arrival rate of the carried traffic is $\lambda(1-P_{bk})$, from Little's results, the expected queueing delay for a call then is

$$E(D_q) = E(N_q)/(\lambda(1-P_{bk})) = \frac{P_{bk}(1-u^h)(1-Su^{S-1}+(S-1)u^S)}{\lambda(1-P_{bk})u^h(1-u)^2(1-u^{S+1})} \tag{6.13}$$

Figures 6.24 shows queueing delay comparison between FACT and the fixed allocation policies. The results show that the queueing delay with FACT is always longer than with conventional policies.

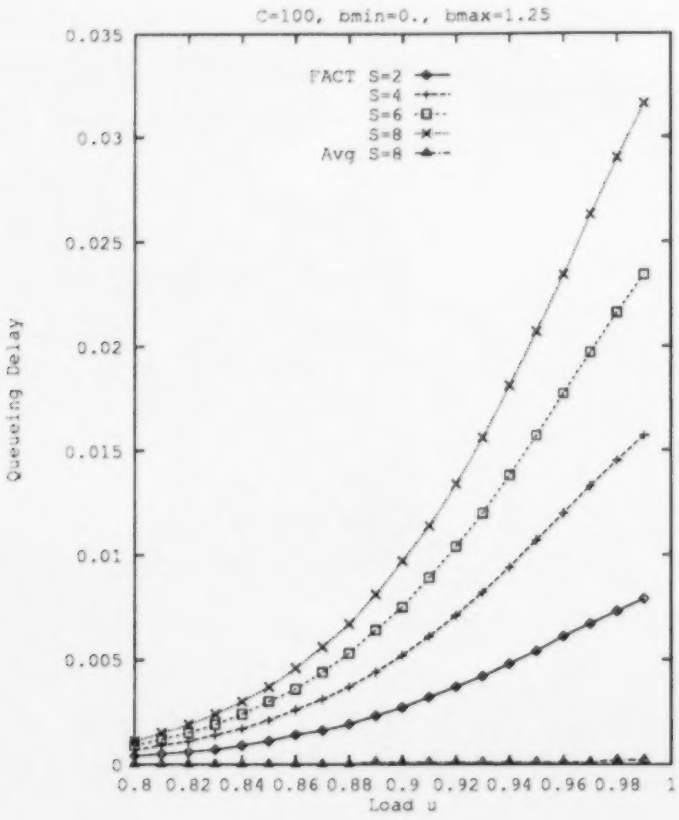

Figure 6.24: Queueing Delay Comparison for Single Flexible Call Class

The results show that the queueing probability increases much faster with FACT than with the fixed allocations. This is a consequence of the different objectives of using the queue in these two types of policies. In the conventional fixed allocation, the queue is used to hold overflow calls only; that is, the calls that cannot be accepted on the VP. The queue in FACT is used mainly for postponing the relinquishment, besides also holding overflow calls. While the high queueing probability with fixed allocation often implies high possibility for blocking, it is not the case with FACT. With FACT, the queueing merely indicates that in-progress calls are using less than the maximum (b_{max}) BUs.

However, the long queueing delay with FACT does not discredit its usefulness yet, since FACT could shorten the call transmission time, and thus, the total call delay. This issue is the topic of the next subsection.

D. Average Call Delay

The *call delay* (D) is defined as the time elapsed from the instant the call request is issued to the instant the call is finished. In other words, call delay is the total time spent by the call in the network, including transmission time and queueing time. The problem of finding the average call delay amounts to finding the average number of calls in the network, because the average delay is the average number of calls divided by $\lambda(1 - P_{bk})$ according to Little's result.

For fixed allocation policies, the expected number of calls in the network can be easily obtained by multiplying the nonblocking probability by the traffic density. The expected number of calls with Maximum Allocation, for example, is $\rho_{max}(1 - P_{bk,max})$.

For the flexible allocation policy, however, it is no longer straightforward to find the expected number of calls and the average call delay, because the bandwidth allocation is no longer constant. The next theorem provides the exact formula for these two measures.

Theorem 6.4 *The average number of calls in the network with FACT is given by*

$$E(k) = \rho_{max}(1 - P_{bk}) + up_L \frac{1 - hu^{h-1} + (h-1)u^h}{(1-u)^2} \tag{6.14}$$

and the corresponding expected call delay is

$$E(D) = \frac{1}{\mu b_{max}} + \frac{1 - hu^{h-1} + (h-1)u^h}{u^h m\mu(1-u)^2} \tag{6.15}$$

Proof. Recall that $\rho_{max} = \rho/b_{max} = C\rho/(b_{max}C) = Lu$ by definition, and $\sum_{i=1}^{n-1} ix^i = \frac{1-nx^{n-1}+(n-1)x^n}{(1-x)^2}x$, thus, the average number of calls in the system is given by

$$\begin{aligned}
E(k) &= \sum_{k=0}^{H+S} kp_k \\
&= \sum_{k=0}^{L} k\rho_{max}^k/k! + \sum_{k=L}^{H+S} kp_k \\
&= \rho_{max}\sum_{k=0}^{L-1} \rho_{max}^k/k! + \sum_{k=L}^{H+S} kp_k \\
&= \rho_{max}\sum_{k=0}^{L-1} p_k + \sum_{k=L}^{H+S} kp_k \\
&= \rho_{max}\sum_{k=0}^{L-1} p_k + p_L\sum_{i=1}^{h}(L+i)u^i \\
&= \rho_{max}\sum_{k=0}^{L-1} p_k + Lp_L\sum_{i=1}^{h} u^i + p_L\sum_{i=1}^{h} iu^i
\end{aligned}$$

$$
\begin{aligned}
&= \rho_{max} \sum_{k=0}^{L-1} p_k + \rho_{max} p_L \sum_{i=1}^{h} u^{i-1} + p_L \sum_{i=1}^{h} i u^i \\
&= \rho_{max} \sum_{k=0}^{L-1} p_k + \rho_{max} \sum_{k=L}^{H+S-1} u^i + p_L \sum_{i=1}^{h} i u^i \\
&= \rho_{max} \sum_{k=0}^{H+S-1} p_k + p_L \sum_{i=1}^{h} i u^i \\
&= \rho_{max}(1 - p_{H+S}) + p_L \sum_{i=1}^{h} i u^i \\
&= \rho_{max}(1 - P_{bk}) + u p_L \frac{1 - h u^{h-1} + (h-1)u^h}{(1-u)^2}
\end{aligned}
$$

Substituting $\rho_{max} = \lambda/(\mu b_{max})$ and $u = \lambda/(m\mu)$, $E(D)$ can then be obtained by applying Little's result.

$$
\begin{aligned}
E(D) &= E(k)/\lambda(1 - Pbk) \\
&= \frac{1}{\mu b_{max}} + \frac{P_{bk}(1 - h u^{h-1} + (h-1)u^h)}{\lambda(1 - P_{bk})(1-u)^2}
\end{aligned}
$$

□

The expected call delay on a VP with FACT always is between that of the maximum allocation policy and the minimum allocation policy, because FACT allocates bandwidth between b_{min} and b_{max}.

The difference in mean call delay between FACT and the fastest fixed allocation – Maximum Allocation – follows from the second term of the summation in (6.15) and the queueing delay with Maximum Allocation $E(D_{q,max})$.

The latter is

$$E(D_{q,max}) = \frac{P_{bk,max}}{\lambda(1 - P_{bk,max})} \sum_{i=1}^{S} u^{-i}$$

where $P_{bk,max}$ is the blocking rate with maximum allocation, and $u_{max} = u/b_{max}$. The former can be expressed as

$$\frac{P_{bk}}{\lambda(1 - P_{bk})} u^{-h} \sum_{i=1}^{h} i u^i = \frac{P_{bk}}{\lambda(1 - P_{bk})} \sum_{i=0}^{h-1} (h-i) u^{-i}$$

According to Theorem 6.1, $P_{bk} \leq P_{bk,max}$, thus,

$$\frac{P_{bk}}{\lambda(1 - P_{bk})} \leq \frac{P_{bk,max}}{\lambda(1 - P_{bk,max})}$$

since $h \leq S + 1$,

$$\sum_{i=0}^{h-1} (h-i) u^{-i} \geq \sum_{i=1}^{S} u^{-i}$$

Therefore, the following upper bound for the call delay with FACT has been proven.

Theorem 6.5 *(a). The call delay with FACT is always shorter than that with the minimum allocation.*
(b). The call delay with FACT is always longer than that with the maximum allocation, and that difference has an upper bound

$$\frac{P_{bk,max}}{1-P_{bk,max}}\left(\sum_{i=0}^{h-1}(h-i)u^{-i}-\sum_{i=1}^{S}u^{-i}\right)$$

Figure 6.25 illustrates call delay values derived from the analysis for various queue lengths. The call delay for the Maximum Allocation policy is close to the horizontal axis at (0.8), due to the negligible queueing delay (shown in Figure 6.24), while it is at 1.0 for the Average Allocation, and at 1.25 for the Minimum Allocation, with slight increases when u increases. Obviously, FACT enjoys a shorter delay than Average Allocation under almost all the situations considered in Figure 6.25.

The next section investigates the impact of the holding queue on the relinquishment frequency.

E. Relinquishment Probability

The relinquishment probability is defined as the likelihood that upon a call arrival in-progress calls are made to relinquish some bandwidth to be used by the new call or a batch of newly admitted calls. It is given in the following theorem.

Theorem 6.6 *The relinquishing probability for a flexible call from a single class is given by*

$$P_{rlq}=\frac{1-u^{h}}{1-u^{S+1}}p_{L+S+1} \tag{6.16}$$

It becomes

$$P_{rlq}=\frac{1-u^{H-L}}{1-u}p_{L+1}, \tag{6.17}$$

if the immediate relinquishment policy is used.

Proof. By definition of the immediate relinquishment policy, relinquishment occurs if, and only if, an arriving call finds between L and $H-1$ calls in progress . Thus, the relinquishing probability with immediate relinquishment policy is

$$\begin{aligned}P_{rlq} &= \sum_{k=L}^{H-1}p_k\\ &= p_L\sum_{k=0}^{H-L-1}u^k\\ &= \frac{1-u^{H-L}}{1-u}p_L\end{aligned}$$

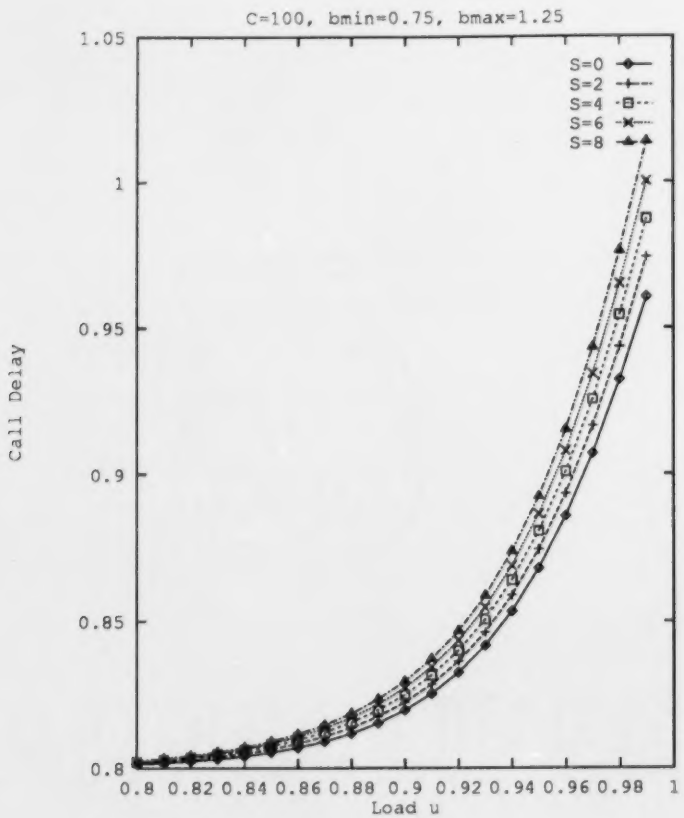

Figure 6.25: Call Delay for Single Class of Flexible Calls

For the delayed relinquishment policy, relinquishment occurs if, and only if, an arriving call finds S waiting calls in the queue, and $N_p < H$ calls in progress. At the time of relinquishment, waiting calls and the new arrival are admitted as a batch from the queue. It takes $s+1$ relinquishments for the system to reach the full state $H+S$ from the empty state. The relinquishment for them occurs at state transitions from $L+S+1$ to $L+S+2$, from $L+2(S+1)$ to $L+2(S+1)+1$,..., and from $L+s(S+1)$ to $L+s(S+1)+1$.

Therefore, the relinquishment probability with the delayed relinquishment policy is

$$\begin{aligned} P_{rlq} &= \sum_{i=1}^{s} p_{L+i(S+1)} \\ &= p_L \sum_{i=1}^{s} u^{i(S+1)} \end{aligned}$$

$$
\begin{aligned}
&= \frac{u^{S+1} - u^{(s+1)(S+1)}}{1 - u^{S+1}} p_L \\
&= \frac{1 - u^{s(S+1)}}{1 - u^{S+1}} p_{L+S+1} \\
&= \frac{1 - u^{h}}{1 - u^{S+1}} p_{L+S+1}
\end{aligned}
$$

□

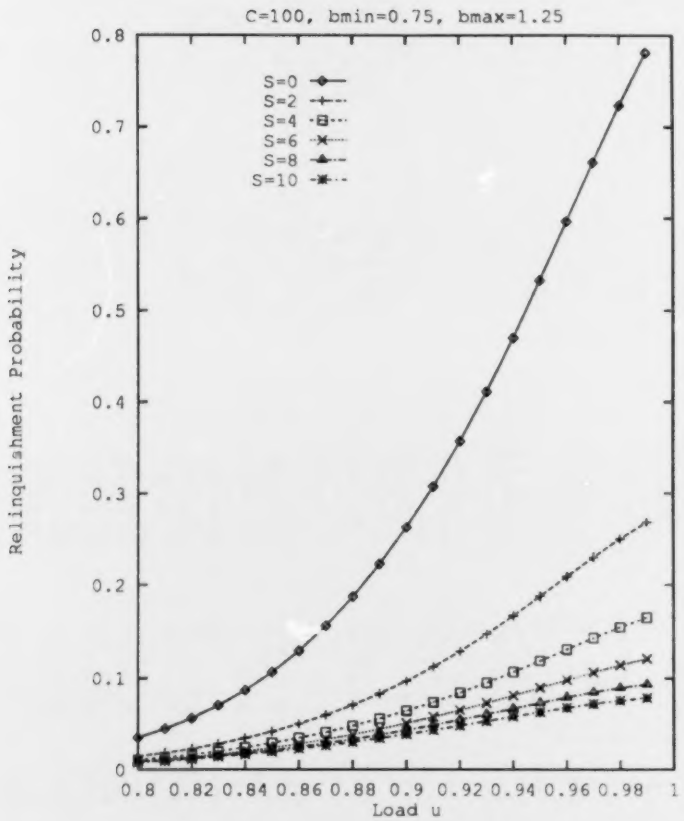

Figure 6.26: Relinquish Probabilities for Single Class of Flexible Calls

Figure 6.26, shows the relinquishment probabilities with the same queue size, capacity and traffic load parameters as in previous examples. The results suggest that the queue does not have to be very long to reduce relinquishment probabilities significantly, as the larger the size of the waiting room S, the less decrease there is in relinquishment probability.

The results also show the effect on the relinquishment probability when a holding queue is used. For instance, when S=0, at very high offered load $u = 0.99$, approximately 8 out of 10 arriving calls will trigger a relinquishment, and at load $u = 0.85$, still 10% of arrivals will cause relinquishment. The relinquishment probability, however, decreases drastically with the delay relinquishment policy. With room for holding only one arrival ($S = 1$), less than 3 out of 10 arrivals will trigger a relinquishment at very high load $u = 0.99$, as compared to 8 out of 10 with $S = 0$. The relinquishment probability generally reduces to more than half with $S = 1$ in the example VP. The results from this example enforce the initial intention of introducing a queue to reduce the relinquishment probabilities.

6.5 Single Flexible Class at the Burst Level

This section considers the same situation as in the previous section, but at the burst level. In other words, all bursts are from the same class, and immediate and delayed relinquishment policies will be used.

6.5.1 Assumptions and Model

The same notations will be followed for the burst level as those for the call/VC level. For example, the state k is the number of bursts (active bursts) in the system. The arrival rate in state k is, however, no longer a constant λ, but $(N - k)\lambda$. The modeled process is also a one-dimensional birth-death process, similar to the call level, as shown in Figure 6.27. The departure rate for each state again is given in Lemma 6.2.

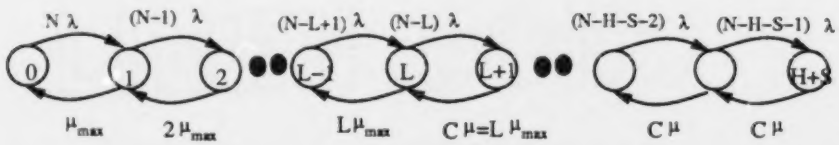

Figure 6.27: State Diagram for a Single Class of Flexible Bursts

The situation of interest here is when $N > L = C/b_{max}$. Otherwise, each burst request can be accommodated with the maximum bandwidth. Moreover, the assumption in Section 6.2.2 of only brief period of overload requires that $(N - k)\lambda/\mu < C$ for $k \geq L$.

The steady state probabilities for both immediate and delayed relinquishment policies can be stated as follows.

Lemma 6.5 *The steady state probability of the system being in state k is given by*

$$p_k = \begin{cases} p_0 \binom{N}{k} \rho_{max}^k & for\ 0 \le k \le L \\ p_0 \binom{N}{k} u^k k! \frac{L^L}{L!} & for\ L < k \le H+S \end{cases}$$

where $\binom{N}{k} = \frac{N!}{k!(N-k)!}$, *and*

$$p_0^{-1} = \sum_{k=0}^{L-1} \binom{N}{k} \rho_{max}^k + \sum_{k=L}^{H+S} \binom{N}{k} u^k k! \frac{L^L}{L!}$$

Proof. Applying Lemma 6.2 to the general birth-death process' state probabilities $p_k = p_0 \prod_{i=0}^{k-1} \lambda_i/\mu_{i+1}$, we have,
for $0 < k \le L$

$$\begin{aligned} p_k &= p_0 \prod_{i=0}^{k-1} \lambda_i/\mu_{i+1} \\ &= p_0 \prod_{i=0}^{k-1} (N-i)\lambda/((i+1)\mu_{max}) \\ &= p_0 \binom{N}{k} \rho_{max}^k \end{aligned}$$

and for $L < k \le H+S$, since $L = C/b_{max}$, and $u = \rho/C$,

$$\begin{aligned} p_k &= p_0 \prod_{i=0}^{L-1} \frac{(N-i)\lambda}{(i+1)\mu_{max}} \prod_{i=L}^{k-1} \frac{(N-i)\lambda}{C\mu} \\ &= p_0 \binom{N}{k} \rho^k \frac{k!}{C^{k-L} b_{max}^L L!} \\ &= p_0 \binom{N}{k} u^k k! \frac{L^L}{L!} \end{aligned}$$

p_0^{-1} immediately follows from $\sum_{k=0}^{H+S} p_k = 1$. □

Once the steady state probabilities have been obtained, the next step is to derive expressions for various performance measures of FACT and compare them with the existing schemes at the burst level. The next section will prove results for the burst level that are similar to those at the VC level.

6.5.2 Performance Measures and Comparisons

A. Burst Blocking Rate

Similar to Lemma 6.4, the probability for a burst from single flexible class traffic of density ρ and population N being blocked at a VP of capacity C with FACT can be obtained from Lemma 6.5 as

$$P_{bk} = p_0 u^{H+S} \binom{N}{H+S} \frac{(H+S)! L^L}{L!} \tag{6.18}$$

The following lemma compares the steady state probabilities of the system being at state 0 for FACT and the fixed allocation policies, where $p_{0,FACT}$ denotes p_0 for FACT in Lemma 6.5 to distinguish it from other policies.

Lemma 6.6 *If* $b_{min} < b_{max}$, *then*

$$\begin{aligned} p_{0,max} &> p_{0,FACT} \\ C^C L! b_{max}^L p_{0,avg} &> C^L C! p_{0,FACT} \\ H^H L! b_{max}^L p_{0,min} &> L^L H! b_{min}^H p_{0,FACT} \text{ if } b_{min} b_{max} \leq 1 \text{ and } \frac{C^L H!}{C^H L!} < 1 \end{aligned}$$

Proof.

By inspecting $p_{0,max}^{-1}$ and $p_{0,FACT}^{-1}$, it is apparent that $p_{0,max}^{-1} < p_{0,FACT}^{-1}$, thus, $p_{0,max} > p_{0,FACT}$

Since $b_{max} > 1$

$$\begin{aligned} b_{max}^L p_{0,FACT}^{-1} &= b_{max}^L \left(\sum_{k=0}^{L-1} \binom{N}{k} \rho_{max}^k + \sum_{k=L}^{H+S} \binom{N}{k} \rho^k \frac{k! L^L}{C^k L!}\right) \\ &> \sum_{k=0}^{L-1} \binom{N}{k} \rho^k + \sum_{k=L}^{H+S} \binom{N}{k} \rho^k \frac{k! C^L}{C^k L!} \\ &> \sum_{k=0}^{L-1} \binom{N}{k} \rho^k + \sum_{k=L}^{C-1} \binom{N}{k} \rho^k \frac{k! C^L}{C^k L!} + \sum_{k=C}^{C+S} \binom{N}{k} \rho^k \frac{k! C^L}{C^k L!} \end{aligned}$$

Since $\frac{L!/C^L}{C!/C^C} > 1$, and $\frac{C^C}{C!} > \frac{C^k}{k!}$ for $k < C$,

$$\begin{aligned} \frac{L! C^{C-L}}{C!} b_{max}^L p_{0,FACT}^{-1} &> \sum_{k=0}^{L-1} \binom{N}{k} \rho^k + \frac{L! C^{C-L}}{C!} \sum_{k=L}^{C-1} \binom{N}{k} \rho^k \frac{k! C^L}{C^k L!} + \sum_{k=C}^{C+S} \binom{N}{k} \rho^k \frac{k! C^L}{C^k L!} \\ &> \sum_{k=0}^{L-1} \binom{N}{k} \rho^k + \sum_{k=L}^{C-1} \binom{N}{k} \rho^k \frac{k! C^C}{C! C^k} + \sum_{k=C}^{C+S} \binom{N}{k} \rho^k \frac{k! C^L}{C^k L!} \\ &> \sum_{k=0}^{L-1} \binom{N}{k} \rho^k + \sum_{k=L}^{C-1} \binom{N}{k} \rho^k / k! + \sum_{k=C}^{C+S} \binom{N}{k} \rho^k \frac{k! C^L}{C^k L!} \\ &= p_{0,avg}^{-1} \end{aligned}$$

Therefore,

$$C^C L! b_{max}^L p_{0,avg} > C^L C! p_{0,FACT}$$

Also, since $\frac{k! C^L}{L! C^k} \geq 1$ when $L \leq k < C$,

$$b_{max}^L p_{0,FACT}^{-1} > \sum_{k=0}^{L-1} \binom{N}{k} \rho^k + \sum_{k=L}^{H+S} \binom{N}{k} u^k k! \frac{C^L}{L!}$$

$$> \sum_{k=0}^{C-1} \binom{N}{k} \rho^k + \sum_{k=C}^{H+S} \binom{N}{k} \rho^k \frac{k!C^L}{C^k L!}$$

$$\begin{aligned}
b_{min}^H p_{0,min}^{-1} &= b_{min}^H \Big(\sum_{k=0}^{H-1} \binom{N}{k} \rho_{min}^k + \sum_{k=H}^{H+S} \binom{N}{k} \rho_{min}^k \frac{k!}{H! H^{k-H}} \Big) \\
&= \sum_{k=0}^{H-1} \binom{N}{k} \rho^k b_{min}^{H-k} + \sum_{k=H}^{H+S} \binom{N}{k} \rho^k \frac{k!C^H}{H!C^k} \\
&= \sum_{k=0}^{L-1} \binom{N}{k} \rho^k b_{min}^{H-k} + \sum_{k=L}^{H-1} \binom{N}{k} \rho^k b_{min}^{H-k} + \sum_{k=H}^{H+S} \binom{N}{k} \rho^k \frac{k!C^H}{H!C^k} \\
&= \sum_{k=0}^{L-1} a_{1,k} + \sum_{k=L}^{H-1} a_{2,k} + \sum_{k=H}^{H+S} a_{3,k} \\
b_{max}^L p_{0,FACT}^{-1} &= b_{max}^L \Big(\sum_{k=0}^{L-1} \binom{N}{k} \rho_{max}^k + \sum_{k=L}^{H+S} \binom{N}{k} \rho^k \frac{k!L^L}{C^k L!} \Big) \\
&= \sum_{k=0}^{L-1} \binom{N}{k} \rho^k b_{max}^{L-k} + \sum_{k=L}^{H+S} \binom{N}{k} \rho^k \frac{k!C^L}{C^k L!} \\
&= \sum_{k=0}^{L-1} \binom{N}{k} \rho^k b_{max}^{L-k} + \sum_{k=L}^{H-1} \binom{N}{k} \rho^k \frac{k!C^L}{C^k L!} + \sum_{k=H}^{H+S} \binom{N}{k} \rho^k \frac{k!C^L}{C^k L!} \\
&= \sum_{k=0}^{L-1} b_{1,k} + \sum_{k=L}^{H-1} b_{2,k} + \sum_{k=H}^{H+S} b_{3,k}
\end{aligned}$$

where notations $a_{i,k}$ and $b_{i,k}$ are used for simplicity.

Because $a_{1,k}/b_{1,k} < 1$, $a_{2,k}/b_{2,k} < b_{min} < b_{max}$, and $a_{3,k}/b_{3,k} = L!C^{H-L}/H!$, the assertion is true if $\frac{H!}{C^{H-L}L!} < 1$ and $b_{min}b_{max} \leq 1$. □

The next theorem proves the advantage of FACT over fixed allocations at the burst level.

Theorem 6.7 *When $b_{min} < b_{max}$,*

(a). FACT always has a lower burst blocking probability on a VP than the policies with the allocation fixed at b_{max} and 1.

(b). Moreover, the burst blocking rate ratios of FACT to these two fixed allocation policies have upper bounds $u^{H-L}(N-L-S)!/(N-H-S)! < 1$ and $u^{H-C}(N-L-S)!/(N-H-S)! < 1$, respectively.

(c). If $H! < L!C^{H-L}$ and $b_{max}b_{min} \leq 1$, a VP with FACT has a lower burst blocking probability than the policy with the allocation fixed at b_{min}.

Proof. Let $B_{max}, B_{avg}, B_{min}$ denote the ratio of the burst blocking probabilities with FACT to that with Maximum, Average and Minimum Allocations, respectively.

First, the proof for the case of fixed allocation at the maximum is as follows. According to Lemma 6.6

$$\begin{aligned} B_{max} &= \frac{p_{0,FACT} u^{H+S} \binom{N}{H+S} \frac{(H+S)!L^L}{L!}}{p_{0,max} \rho_{max}^{L+S} \binom{N}{L+S} \frac{(L+S)!}{L!L^S}} \\ &< \frac{u^{H+S} \binom{N}{H+S} \frac{(H+S)!L^L}{L!}}{\rho_{max}^{L+S} \binom{N}{L+S} \frac{(L+S)!}{L!L^S}} \\ &= \frac{u^{H+S} \binom{N}{H+S} \frac{(H+S)!L^L}{L!}}{u^{L+S} \binom{N}{L+S} \frac{(L+S)!L^L}{L!}} \\ &= u^{H-L} \frac{(N-L-S)!}{(N-H-S)!} \end{aligned}$$

Because $(N-k)u < 1$ for $k \geq L$ as a result of the temporary network overload condition, $u/(N-k) < 1$ for $k > L+S$. Therefore, $B_{max} < 1$.

A similar result can be derived for the Average Allocation.

$$\begin{aligned} B_{avg} &= \frac{p_{0,FACT} u^{H+S} \binom{N}{H+S} (H+S)! \frac{L^L}{L!}}{p_{0,avg} \rho^{C+S} \binom{N}{C+S} \frac{(C+S)!}{C!}} \\ &< \frac{C^C L! b_{max}^L u^{H+S} \binom{N}{H+S} (H+S)! \frac{L^L}{L!}}{\rho^{C+S} \binom{N}{C+S} \frac{(C+S)!}{C!}} \\ &= \frac{C^L C! u^{H+S} \binom{N}{H+S} (H+S)! \frac{C^L}{L!}}{u^{C+S} \binom{N}{C+S} (C+S)! \frac{C^C}{C!}} \\ &= \frac{u^{H+S} \binom{N}{H+S} (H+S)!}{u^{L+S} \binom{N}{L+S} (L+S)!} \\ &= u^{H-C} \frac{(N-C-S)!}{(N-H-S)!} \\ &< 1 \end{aligned}$$

An upper bound for the ratio of blocking probabilities with FACT to the Minimum Allocation

can be obtained, under the conditions that $b_{min}b_{max} \leq 1$ and $H! < C^{H-L}L!$,

$$\begin{aligned}
B_{min} &= \frac{p_{0,FACT}u^{H+S}\left(H+S\atop H\right)(H+S)!\frac{L^L}{L!}}{p_{0,min}\rho_{min}^{H+S}\left(H+S\atop H\right)\frac{(H+S)!}{H!H^S}} \\
&= \frac{p_{0,FACT}u^{H+S}\frac{L^L}{L!}}{p_{0,min}\rho_{min}^{H+S}\frac{1}{H!H^S}} \\
&= \frac{p_{0,FACT}u^{H+S}\frac{L^L}{L!}}{p_{0,min}\rho^{H+S}\frac{1}{H!H^S b_{min}^{H+S}}} \\
&= \frac{p_{0,FACT}u^{H+S}\frac{L^L}{L!}}{p_{0,min}\rho^{H+S}\frac{H^H}{H!C^{H+S}}} \\
&= \frac{p_{0,FACT}L^L H!}{p_{0,min}H^H L!} \\
&= \frac{H!}{L!C^{H-L}}\frac{b_{min}^H p_{0,FACT}}{b_{max}^L p_{0,min}} \\
&< 1
\end{aligned}$$

□

This theorem is an extension of Theorem 6.1 from the infinite population to the finite population case, or from the VC level to the burst level. Thus the same conclusions can be expected to hold for bursts as well as for calls. Similarly, it can be shown that FACT results in higher utilization at the burst level than do the fixed allocation policies.

All other performance measures can also be obtained from the probability distribution in a fashion similar to that for the VC level, and hence they are not repeated here.

6.6 Mixed Traffic of Fixed and Flexible Calls

This section considers a network serving traffic of mixed types; that is, flexible calls served with FACT and conventional calls with fixed bandwidth allocations. In other words, the traffic here is considered to be a coexistence of the new flexible traffic and the existing traffic. This problem is interesting because it indicates how the network will react if the new service class is introduced to existing networks. The other interesting application is to consider how much of the existing traffic should sign up for the new flexible service, and what the consequences are.

The early relinquishment policy defined in Chapter 4 is used for mixed traffic.

This section considers the performance issues only for bandwidth allocation. Another important issue is optimal access control. This will be the topic of the next chapter.

Traffic on a VP will be modeled as a stochastic process, and the equilibrium probabilities will be derived in product forms. Then, various performance measures of interest are obtained and compared to those of the conventional policies. The discussion will be based on the VC level only. The study can easily be extended to the burst level by replacing fixed rate λ_2 with $(N_2 - n_2)\lambda_2$.

6.6.1 Analytical Model

Denote the number of fixed and flexible calls in progress by n_1 and n_2 respectively; and let the row vector $\mathbf{n}$ be the state of the stochastic system of interest, where $\mathbf{n} = (n_1, n_2)$. Notations are: $\mathbf{n_1^-} = (n_1 - 1, n_2)$, $\mathbf{n_1^+} = (n_1 + 1, n_2)$, $\mathbf{n_2^-} = (n_1, n_2 - 1)$, $\mathbf{n_2^+} = (n_1, n_2 + 1)$, and column vectors $\mathbf{b_{max}} = \begin{pmatrix} 1 \\ b_{max} \end{pmatrix}$, $\mathbf{b_{min}} = \begin{pmatrix} 1 \\ b_{min} \end{pmatrix}$, and $\mathbf{b} = \begin{pmatrix} 1 \\ b_2 \end{pmatrix}$ as the current bandwidth allocation for the two classes.

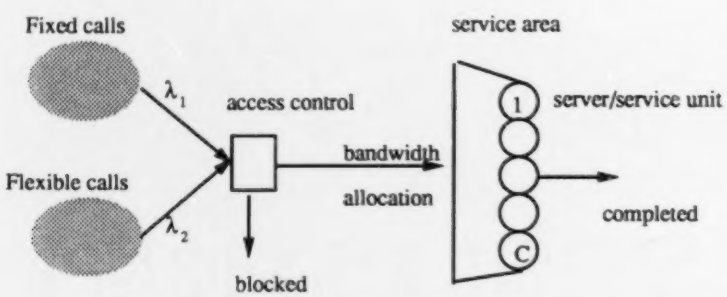

Figure 6.28: Model for Mixed Traffic

The communication network serving both fixed and flexible traffic is depicted in Figure 6.28. The system behavior can then be described as follows.

Call Arrival and Departure Processes

- Calls arrive independently from the two classes, fixed and flexible, according to Poisson processes from infinite population with mean arrival rates λ_1 and λ_2, respectively.
- The call lengths are exponentially distributed for both classes with mean $1/\mu_1$ and $1/\mu_2$, respectively.
- The fixed calls are to be served each by exactly one BU. The flexible calls can be served by any real number of bandwidth between b_{min} and b_{max} BUs inclusively, where $b_{min} \leq 1 \leq b_{max}$.
- An arriving class-i call is accepted at state $\mathbf{n}$ if the new state $\mathbf{n_i^+}$ still satisfies admission conditions.

- The bandwidth allocated to each fixed call is always 1 BU for the duration of the call. The bandwidth allocated to in-progress flexible calls, b, is a function of their number, n_2, so that when n_2 changes due to flexible call arrivals and departures, b is also adjusted by bandwidth relinquishment or re-allocation.

$$b = \begin{cases} b_{max} & if n_2 \leq L = \lfloor (C-m)/b_{max} \rfloor \\ 1 & if L < n_2 \leq C - m \\ b_{min} & if C - m < n_2 \leq H = \lfloor C/b_{min} \rfloor \end{cases} \tag{6.19}$$

where m is such that $C - m$ is one of the early relinquishment thresholds as defined in Section 4.2.2. The other threshold is L. When the number of in-progress flexible calls reaches L or $C - m$, bandwidth relinquishment is induced so that, respectively, either 1 or b_{min} BU are allocated to each in-progress flexible call.

The modeled process for mixed traffic on a VP is a two dimensional birth-death process. Let $\mu_1(\mathbf{n})$ and $\mu_2(\mathbf{n})$ be the transition rates from state $\mathbf{n}$ to states $\mathbf{n_1^-}$ and $\mathbf{n_2^-}$, respectively. Since the departure rate of fixed calls is independent of flexible calls, we have

$$\mu_1(\mathbf{n}) = n_1 \mu_1 \tag{6.20}$$

Using an approach similar to that for a single class in the previous section, we obtain

$$\mu_2(\mathbf{n}) = \begin{cases} n_2 b_{max} \mu_2 & if\ n_2 \leq L \\ n_2 \mu_2 & if\ L < n_2 \leq C - m \\ n_2 b_{min} \mu_2 & if\ C - m < n_2 \leq H \end{cases} \tag{6.21}$$

Probability Distribution for Carried Mixed Traffic

A state $\mathbf{n}$ is said to exist if n can be reached by FACT. Let $p(\mathbf{n})$ denote the steady state probability for state $\mathbf{n}$. The flow balance equation is given by

$$\sum_{i=1}^{2} (\lambda_i(\mathbf{n}) + \mu_i(\mathbf{n}))p(\mathbf{n}) = \sum_{i=1}^{2} (\mu_i(\mathbf{n_i^+})p(\mathbf{n_i^+}) + \lambda_i(\mathbf{n_i^-})p(v_i^-)) \tag{6.22}$$

if all the states mentioned in the equation exist. If one or more states do not exist, then the corresponding terms at both sides of the equation disappear. For instance, if state $\mathbf{n_2^+}$ exists, but not $\mathbf{n_1^+}$, then only the first terms on both sides disappear. The local balance equations are

$$\lambda_i p(\mathbf{n}) = \mu_i(\mathbf{n_i^+})p(\mathbf{n_i^+}) \tag{6.23}$$

where $i = 1, 2$.

For $i = 2$,(i.e., the flexible class) and when $n_2 \leq L$, applying (6.21) the local balance equation (6.23) becomes

$$\lambda_2(n_2)p(\mathbf{n}) = (n_2 + 1)b_{max}\mu_2 p(\mathbf{n_2^+})$$

and, when $L < n_2 \leq C - m$, it becomes

$$\lambda_2(n_2)p(\mathbf{n}) = n_2\mu_2 p(\mathbf{n}_2^+)$$

The state transition diagrams are given in Figure 6.29 for the three regions.

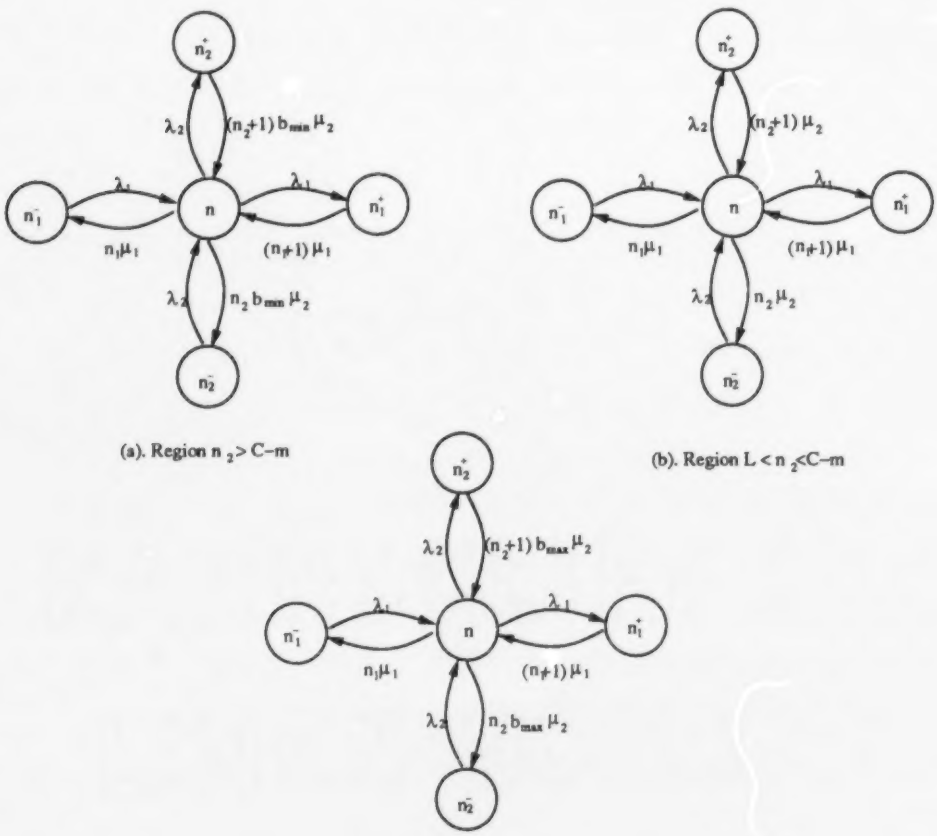

Figure 6.29: State Transitions for Mixed Calls

The steady state probabilities can then be found as

$$p(\mathbf{n}) = q_1(n_1)q_2(n_2)p(\mathbf{0}) \tag{6.24}$$

with $p(\mathbf{0})$ as the normalization factor such that $\sum p(\mathbf{n}) = 1$, and

$$q_1(n_1) = \frac{\rho_1^{n_1}}{n_1!} \tag{6.25}$$

$$q_2(n_2) = \begin{cases} \rho_2^{n_2}/(b_{max}^{n_2} n_2!) & for\ n_2 \le L \\ \rho_2^{n_2}/(n_2! b_{max}^{L}) & for\ L < n_2 \le C - m \\ \rho_2^{n_2}/(n_2! b_{min}^{n2-C+m} b_{max}^{L}) & for\ C - m < n_2 \le H(0) \end{cases} \tag{6.26}$$

The proof of that theorem is obtained by verifying in (6.22).

6.6.2 Performance Measures and Comparisons

Once the steady state probabilities have been obtained, we then proceed to derive various performance measures, such as utilization, average delay, blocking rate and so on. Due to the complexity of the product form for the state probabilities, these measures can hardly be expressed in a compact form. We are nevertheless able to provide the necessary formulas to facilitate the calculation of various performance measures from the steady state probabilities. The process to obtain the performance measures numerically is fairly straightforward and the computation cost is moderate. The complexity in fact is $O(C^2)$, a polynominal function of the capacity C.

Call Blocking Rate

Let function $H(x)$ be the maximal number of flexible calls that can be present in the system when there are x fixed calls, and let function $h(y)$ be the maximal number of fixed calls that can be present in the the system when there are y flexible calls in the system. Thus,

$$H(x) = \begin{cases} \lfloor (C - x)/b_{min} \rfloor & if\ \ x \le m \\ \lfloor (C - x)/b_{max} \rfloor & if\ \ x > m \end{cases}$$

$$h(y) = \begin{cases} \lfloor C - y b_{min} \rfloor & if\ \ y \ge C - m \\ m & if\ \ L \le y \le C - m \\ \lfloor C - y b_{max} \rfloor & if\ \ 0 \le y < L \end{cases}$$

Theorem 6.8 *If $L = H$, then flexible and fixed calls have the same blocking rate. If $L < H$, then the call blocking probability for the flexible class is always smaller than that of the fixed class, regardless of their respective arrival rates and lengths. Moreover, the blocking rate difference is more than*

$$\frac{\rho_1^m (E_H(\rho_2) - E_{L-1}(\rho_2))}{m b_{max}^L} p(0)$$

Proof. Let P_{bki} denote the call blocking rate for class-i (i=1,2), then we have

$$P_{bk1} = \sum_{x=0}^{C} p(x, H(x)) + \sum_{y=L}^{H-1} p(m, y) \tag{6.27}$$

$$P_{bk2} = \sum_{x=0}^{C} p(x, H(x)) \tag{6.28}$$

When $H > L$, $P_{bk1} > P_{bk2}$ because $P_{bk1} - P_{bk2} = \sum_{y=L}^{H-1} p(m, y) > 0$, regardless of $\rho_1, \rho_2, m, b_{min}$ or b_{max}.

$$
\begin{aligned}
P_{bk1} - P_{bk2} &= \sum_{y=L}^{H-1} p(m,y) \\
&= q_1(m) \sum_{y=L}^{H-1} q_2(y) \\
&= q_1(m)(q_2(L) + \sum_{y=L+1}^{C-m} \frac{\rho_2^y}{y! b_{max}^L} p(\mathbf{0}) + \sum_{y=C-m+1}^{H-1} \frac{\rho_2^y}{y! b_{min}^{y-H} b_{max}^L} p(\mathbf{0})) \\
&> q_1(m) \sum_{y=L}^{H-1} \frac{\rho_2^y}{y! b_{max}^L} p(\mathbf{0}) \\
&= q_1(m) \sum_{y=L}^{H-1} \frac{\rho_2^y}{y! b_{max}^L} p(\mathbf{0}) \\
&= q_1(m) \frac{\rho_1^m (E_H(\rho_2) - E_{L-1}(\rho_2))}{m b_{max}^L} p(\mathbf{0})
\end{aligned}
$$

□

This theorem implies that the call blocking rate can always be reduced by changing some traffic from fixed to flexible.

Figure 6.30 illustrates this point by showing the call blocking probabilities for both fixed and flexible calls, with *mix* as the percentage of calls converted to flexible from the original all fixed calls for the same offered traffic load u.

The results show that the blocking rate difference between these two classes becomes insignificant when the amount of fixed calls is very small compared to the flexible calls, when the threshold m is kept constant. The explanation is that when the number of the fixed call is far below $m = C/2$, then fixed calls' blocking rate is not affected by flexible calls. If the value of m is changed in relation to the amount of fixed call traffic, then the blocking rate gap should be relatively stable. This is confirmed by the results shown in Figure 6.31, where m is set equal to be $(1 - mix)C$, *mix* is the traffic mixture measured by the bandwidth utilization ratio of flexible calls to fixed calls.

The final question is whether improvement in the call blocking rate is cost effective by providing this new flexible service.

Average Call Delay

The marginal distribution $p_i(n_i)$ can be obtained as follows.

$$
p_1(n_1) = [E_L(\rho_2/b_{max}) + \frac{\rho_2^L (u - u^{H-L+1})}{b_{max}^L (1-u) L!}] \frac{\rho_1^{n_1}}{n_1!} \tag{6.29}
$$

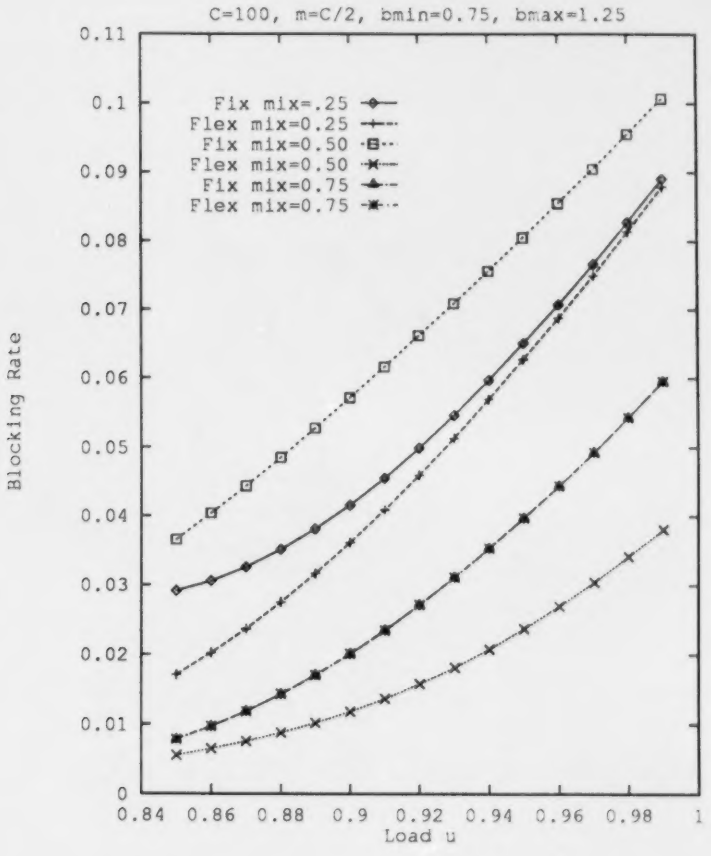

Figure 6.30: Blocking Rates for Mixed Calls with Constant m

$$p_2(n_2) = \frac{\rho_2^{n_2}}{n_2! b_{max}^L} E_{m-n_2 b_{max}} \rho_1 + \rho_2^{n_2} \sum_{\mathbf{nb_{max}}=m+1}^{\mathbf{nb_{min}}=C} \frac{\rho_1^{n_1}}{n_1! b_{max}^L (C-n_1)^{n_2-L} L!} \tag{6.30}$$

The average number of fixed calls in progress is given by

$$E(n_1) = \sum_{i=0}^{C} i p_1(i) \tag{6.31}$$

The average number of flexible calls in progress is given by

$$E(n_2) = \sum_{j=0}^{C/b_{min}} j p_2(j) \tag{6.32}$$

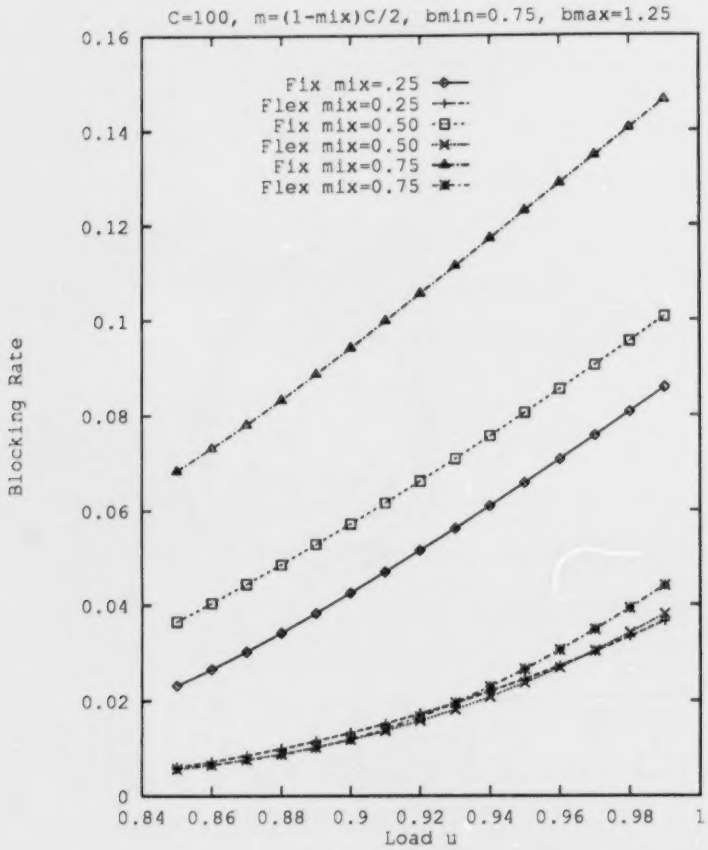

Figure 6.31: Blocking Rates for Mixed Calls with Variable m

From Little's results, we can obtain immediately that the expected call delay for a fixed call is

$$E(D_1) = \frac{E(n_1)}{\lambda_1(1 - P_{bk1})} \tag{6.33}$$

and the expected call delay for flexible call is

$$E(D_2) = \frac{E(n_2)}{\lambda_2(1 - P_{bk2})} \tag{6.34}$$

The maximum and minimum delay for a flexible call are 1.33 and 0.8, when the bandwidth allocation is at the minimum and maximum, respectively, for the call duration. Figure 6.32 shows the flexible call delay for different traffic mixtures when $m = C/2$ is constant. The following explains why the call delays are close to constant at the minimum and maximum for low (25%) and high

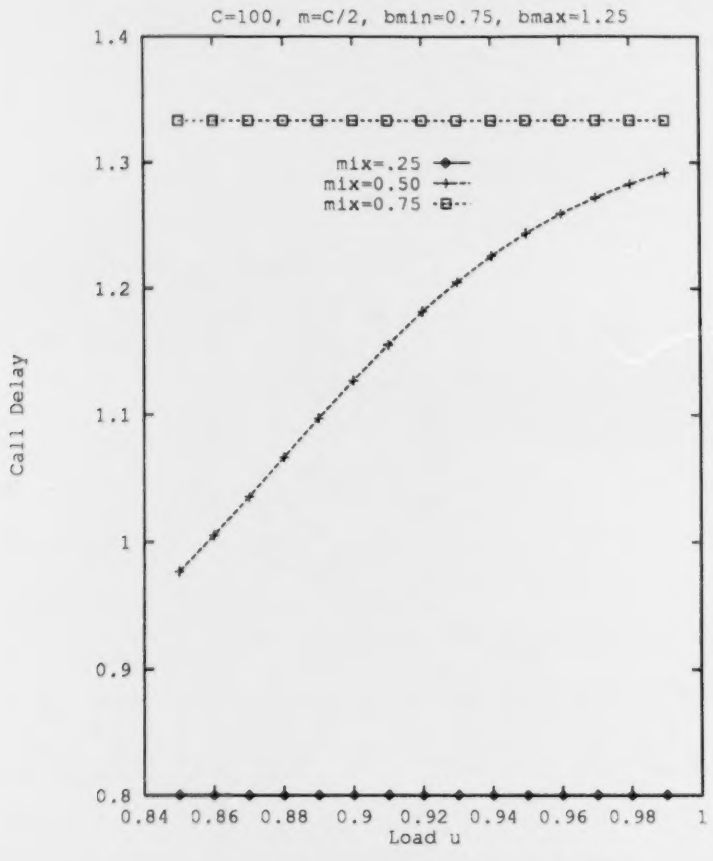

Figure 6.32: Call Delay for Mixed Traffic with Fixed m

(75%) flexible traffic mixtures, respectively. When there are few flexible calls, each flexible call gets the maximum bandwidth allocation, because m is the relatively small compared to the expected number of fixed calls in progress. When m becomes relatively small as the number of flexible calls increase, on the other hand, the bandwidth contention intensifies among flexible calls, and each flexible call is allocated its minimum bandwidth by FACT.

Similarly for the blocking rate, one would guess that the bandwidth contention among flexible calls will stay relatively stable with the threshold m changing with the traffic mixture. The situation for variable $m = (1-mix)C$ shown in Figure 6.33 confirms this guess. By tuning the relation between m and mix, it seems that any desired average flexible call delay can be obtained within the range of $1/b_{min} = 1.33$ and $1/b_{max} = 0.8$, while the call delay for fixed calls is always 1.

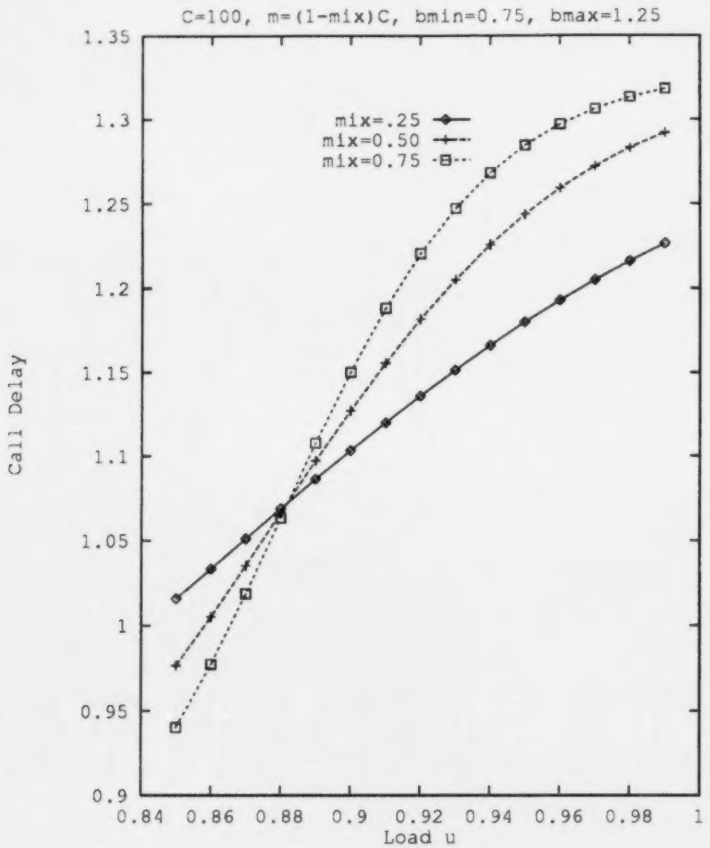

Figure 6.33: Call Delay for Mixed Traffic with Variable m

Bandwidth Utilization

The exact formula for bandwidth utilization is given by

$$E(u) = \sum_{n1<C} \sum_{n_2 \le L} \mathbf{nb_{max}} p(\mathbf{n}) + \sum_{n_1<m} \sum_{L<n_2 \le C-m} (n_1+n_2)p(\mathbf{n}) + \sum_{n_1<m} \sum_{C-m<n_2 \le H(n_1)} \mathbf{nb_{min}} p(\mathbf{n}) \tag{6.35}$$

Analogous to the single class traffic situation, one might, following Theorem 6.8, venture to conjecture that bandwidth utilization can be improved by converting fixed calls into flexible calls. Figure 6.34 shows utilization for variable m. The interesting result is that expected bandwidth utilization is at its lowest when there are 40% of flexible calls among the offered traffic, and it exceeds all the previous values when this ratio (mix) is increased to 60%, and will keep growing with

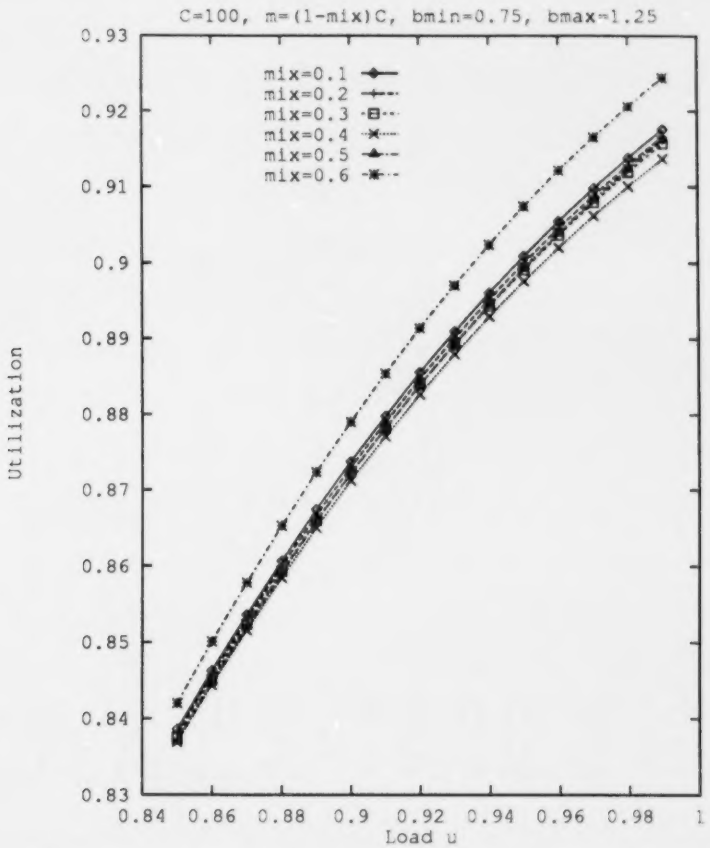

Figure 6.34: Bandwidth Utilization for Mixed Traffic with Variable m

increasing the mix percentage. This result indicates that the bandwidth utilization does not always improve by converting more fixed calls into flexible calls.

Relinquishment Probability

A relinquishment happens only when the state transition moves up vertically crossing the region boundary in the state diagram. In other words, the arrival request triggers relinquishment when the current state is $n_2 = L$ or $n_2 = C - m$. Thus, the relinquishment probability is given by

$$P(relinq) = p_2(L) + p_2(C - m) \tag{6.36}$$

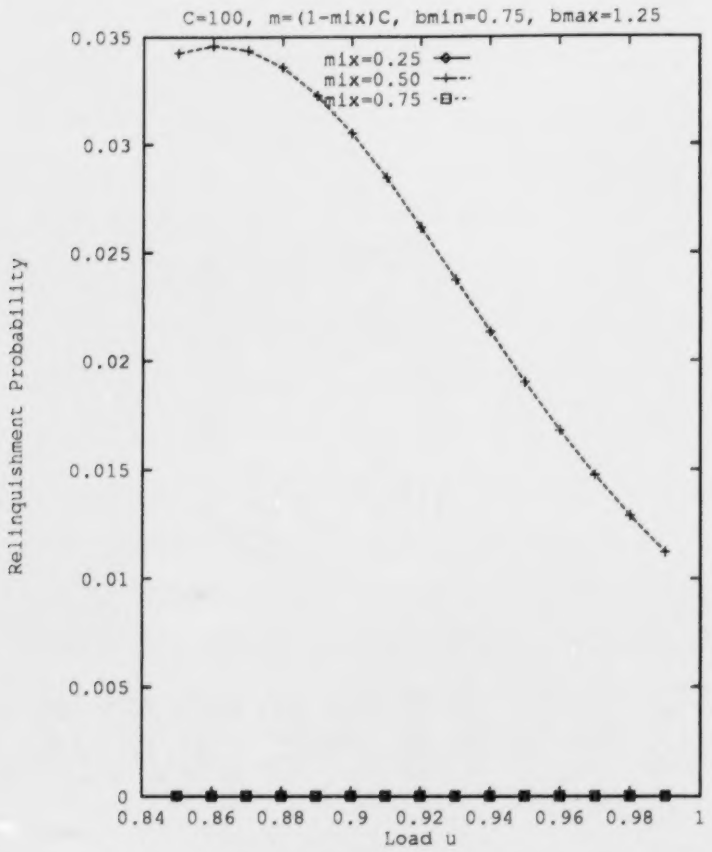

Figure 6.35: Relinquishment Probability for Mixed Traffic with Variable m

Figure 6.35 shows the relinquishment probabilities for mixed traffic with $m = (1 - mix)C$. It agrees with the results of call delay in that the flexible calls are allocated the minimum bandwidth when their number is relatively large, therefore allowing little relinquishment activity.

So far, traffic of one class of flexible calls mixed with traffic from one class of fixed calls has been studied. The analysis can be extended to multiple classes of fixed call traffic mixed with one class of flexible call traffic without substantial difficulty. The extension beyond one class of flexible calls, however, remains an open problem. Several attempts have been made and some experiences gained. The next section discusses these attempts and discoveries.

6.7 Multiple Classes

Three attempts have been made to obtain analytical solutions for traffic with multiple flexible classes. They are the following.

1. Complete partitioning, where call acceptance, bandwidth relinquishment and re-allocation are all done within the same class only.
2. Virtual partitioning, where relinquishment and re-allocation are made within the same class only, and the acceptance of a new arrival is based on bandwidth availability in the VP regardless of classes.
3. Complete sharing, where relinquishment and re-allocation apply to all classes, and no partitioning of bandwidth resources is made at all.

The difference between a virtual partitioning and a complete partitioning is that, in the latter, a new arrival will be accepted only if there is available bandwidth for the arriving call's class. The first two attempts to complete the analysis were unsuccessful, and the difficulties are analyzed and documented here. For the last attempt, complete partitioning, the analysis has been carried out. Here, these attempts and experiences are described for two flexible classes.

Again, it is assumed that class-i call requests arrive independently according to Poisson arrival processes with rate λ_i. The amount of information to be transmitted for a class-i call is assumed to be exponentially distributed with mean $1/\mu_i$. The bandwidth of a class-i call can be any value between $b_{min,i}$ and $b_{max,i}$.

Attempt 1: Complete Partitioning

Complete partitioning is a classic resource sharing strategy. It partitions the bandwidth (or resource in general) among all classes, and calls from the same class deal with the resource dedicated to that class only. Thus the bandwidth allocation and access control for each class is independent of the other classes.

The novelty in our study is to allow that each class be flexible. Because each class behaves independently with the complete partitioning policy, the single class solution can then be applied to each partition. Therefore, the solution to complete partitioning is already known from the study in the previous section. There does, however, exist a new problem for multiple flexible CBR traffic classes with complete partitioning. That is, how to partition the bandwidth among all the classes. It is a practical design issue depending on the particular traffic and network parameters and various performance requirements. Given these parameters and requirements, the analytical results in this

study can be used to find the optimal partitioning. An example procedure is discussed in Chapter 8: Implementation and Application Issues.

Attempt 2: Virtual Partitioning

By virtual partitioning we mean that re-allocation and relinquishment apply to the same class only, whereas the idle capacity can be given to new arrivals from any class. This policy will then result in the "death rate" for state (x, y) being different, perceivably easier than that in Attempt 1 because only one class is involved. However, the difficulty arises from the fact that the process can no longer be characterized by the number of calls (x, y). This is because the process have different characteristics even with the same (x, y) if that state is reached from different routes. For example, $(1, L-1)$ may mean either that class-1 uses its maximum bandwidth and then class-2 calls are assigned equally with less than their maximal requirement, or that class-2 calls are accepted first at their maximal requirement, and then the class-1 calls are accepted at whatever capacity left. These two routes have different departure rates, and cannot be characterized as the same state. So, the virtual partitioning will have a very large state space and is not attractive from the computation complexity view point.

Attempt 3: Complete Sharing

Figure 6.36 illustrates the state diagram for two classes of flexible calls. The number of calls is denoted by x for the first class, and y for the second class. It is assumed that the bandwidth requirement of the second class is always b times of that of the first class calls. The maximal number of the second class calls that can be served by capacity C is therefore H/b, where H is the maximal number of the first class calls that can be served by C.

If $0 < x + by \leq L$, i.e., in zone-I of Figure 6.36,

$$\begin{aligned} & (\lambda_{1,max} + \lambda_{2,max} + \mu_{1,max}x + \mu_{2,max}y)p(x,y) \\ = \; & \lambda_{1,max}p(x-1,y) + \mu_{1,max}(x+1)p(x+1,y) \\ & + \lambda_{2,max}p(x,y+1) + \mu_{2,max}(y+1)p(x,y+1) \end{aligned}$$

If $L \leq x + by \leq H$, i.e., in zone-II in Figure 6.36,

$$\begin{aligned} (\lambda_1 + \lambda_2 + \mu_1 x + \mu_2 y)p(x,y) \; = \; & \lambda_1 p(x-1,y) + \mu_1(x+1)p(x+1,y) \\ & + \lambda_2 p(x,y+1) + \mu_2(y+1)p(x,y+1) \end{aligned}$$

And if $L - 1 \leq x + by \leq L$ and $L - b \leq x + by \leq L$, i.e., both states $(x+1, y)$ and $(x, y+1)$ are outside zone-I,

$$(\lambda_{1,max} + \lambda_{2,max} + \mu_{1,max}x + \mu_{2,max}y)p(x,y)$$

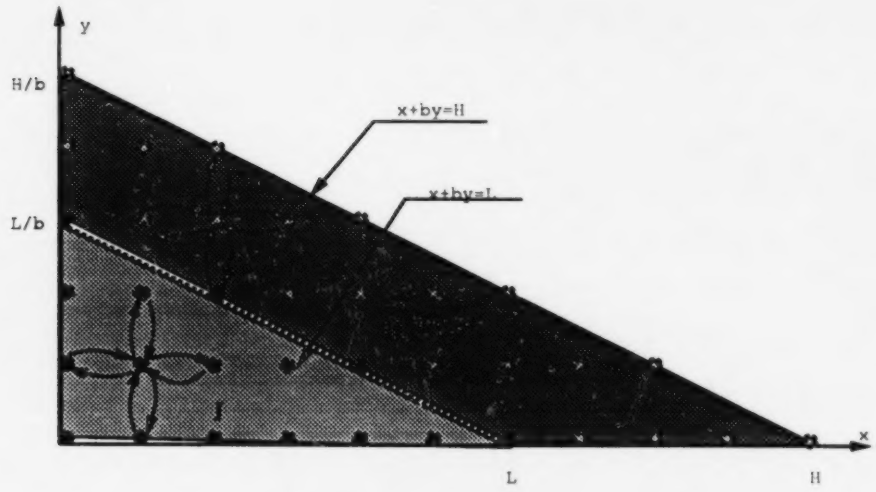

Figure 6.36: State Diagram for Two Flexible Classes

$$= \lambda_{1,max}p(x-1,y) + \mu_{1,max}(x+1)p(x+1,y)$$
$$+\lambda_{2,max}p(x,y+1) + \mu_{2,max}(y+1)p(x,y+1)$$

And if $L-1 \leq x+by \leq L$ and $L-b \geq x+by$, i.e., only $(x+1,y)$ is outside zone-I, And if $L-1 \geq x+by$ and $L-b \leq x+by \leq L$, i.e., only $(x,y+1)$ is outside zone-I,

At this point, it seems that the state probability formula cannot be derived in a product form. Nevertheless, the performance measures can still be obtained numerically.

6.8 Summary

In this chapter, the performance issues of dynamic bandwidth allocation have been studied for FACT.

When a VP carries traffic from the same class, this thesis has proved that FACT can achieve better average blocking and utilization than the allocation fixed at any non-minimum bandwidth requirement for both call and burst levels. The maximum allocation may achieve shorter call delay than FACT. This thesis, has found an upper limit for the average call delay with FACT.

For the policy with allocation fixed at the minimum, this thesis has found certain conditions under which it is outperformed by FACT in call/burst blocking and bandwidth utilization. Moreover, FACT can always result in shorter call delay than the fixed minimum allocation policy.

This chapter also identified that increasing the range of acceptable bandwidth is more effective in improving blocking and utilization than providing additional queue capacity.

It is proved in this thesis that the call blocking rate can always be reduced by converting the existing fixed bandwidth calls into flexible bandwidth calls. The conversion, however, may not always improve the utilization, depending on the choice of boundary for the fixed call m. If m is constant, then the utilization can always be improved by converting more fixed calls into flexible ones. If m is a function of the mixture of the traffic, the utilization, however, is not necessarily improved by the conversion.

For multiple flexible call classes, the same results are applicable, if complete partitioning is used for a VP. Otherwise, results are still to be found.

Chapter 7

Dynamic Access Control: Optimal Policies

7.1 Overview

When there is more than one class of calls, access control policies can be employed to determine for each state whether or not an arriving call should be accepted by assigning a decision – admission or blocking – to each possible state transition. The state is normally the number of in-progress calls of each class. Given a set of states, a policy can be found among all possible access control policies such that the reward is optimized by applying this policy. This policy is called an optimal access control policy. The reward measure varies according to optimization objectives. For network service providers, the reward usually is the profit, i.e, the revenue based on tariff and usage minus the cost of providing the services. The optimal access control policy hence, in this case, is to maximize the profit over a given time period which can be just the instant of the decision, a infinite time period, or any period between these two.

The static access control studied in Chapter 5 is concerned with instantaneous performance. Whereas the dynamic access control considered in this chapter is interested in statistical performance over a long period of time, considering traffic as stochastic process with random arrivals and departures of calls and bursts.

The main results in this chapter deal with mixed traffic of two classes: one of fixed bandwidth calls and one of flexible bandwidth calls. An analytical model for mixed classes is presented in the next section. It is an extension of the model for bandwidth allocation of mixed traffic in Chapter 6. This extension is applicable for not only the VC level but also the burst level. The discussions of access control policies are limited to a class of policies called coordinate convex policies. The optimal policies are proved in Section 7.3 to have a certain structure, namely, a threshold structure.

The difficulties and possible directions for optimal control for more than two classes are discussed in the last section.

7.2 Model for Mixed Classes of Fixed and Flexible Calls

This section considers a network serving traffic that is a mixture of flexible and fixed type calls. The bandwidth allocation problems have been considered in the previous chapter for the VC level control. The stochastic system behavior was described there for call handling. The same model can be expanded to represent both the VC level and the burst level by changing the constant arrival rate λ_i at state $\mathbf{n}$ to a state-dependent non-increasing $\lambda_i(n_i)$. When $\lambda_i(n_i) = \lambda_i$, the traffic is said to be Erlang; and when $\lambda_i(n_i) = (N_i - n_i)\lambda_i$, the traffic is said to be Engset. The VC level traffic is Erlang, because of the infinite subscribers assumption, and the burst level traffic is Engset because the population is a finite number, the number of in-progress calls (N_i). Figure 7.37 depicts the unified model for both levels, where the arrival rates from both classes are the generalized $\lambda_i(n_i)$ with n_i being the number of class-i in-progress calls. The access control part decides whether or not to accept the arriving calls according to the optimal access control policy.

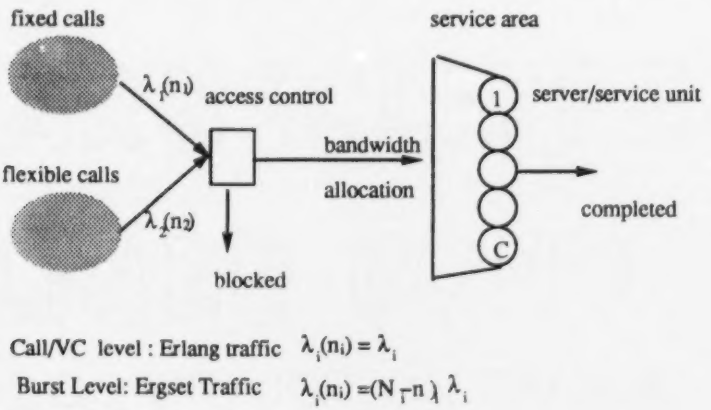

Figure 7.37: Unified Model for the Call and Burst Levels with Mixed Traffic

Similar to the model in Section 6.5, the set of all possible states is:

$$\Omega_F = \{\mathbf{n} : (\mathbf{nb_{min}} \leq C) \wedge (n_1 \leq m \vee \mathbf{nb_{max}} \leq C)\}$$

Access control policies operate on a subset of Ω_F.

The state transition diagram for the unified model is the same as Figure 6.29, except that the arrival rate λ_i is now replaced by a generalized $\lambda_i(n_i)$.

Similarly, the steady state probability distribution can be found as

$$p(\mathbf{n}) = q_1(n_1)q_2(n_2)p(\mathbf{0}) \tag{7.1}$$

with $p(\mathbf{0})$ as a normalization factor such that $\sum_{\mathbf{n}\in\Omega_F} p(\mathbf{n}) = 1$, but the factors q_1 and q_2 are based on the generalized $\lambda_i(n_i)$ as

$$q_1(n_1) = \frac{\prod_{i=0}^{n_1-1}\rho_1(i)}{n_1!} \tag{7.2}$$

$$q_2(n_2) = \begin{cases} \dfrac{\prod_{j=0}^{n_2-1}\rho_2(j)}{b_{min}^{n_2}n_2!} & for\ n_2 \le L \\ \dfrac{\prod_{j=0}^{n_2-1}\rho_2(j)}{n_2!b_{max}^{L}} & for\ L < n_2 \le C-m \\ \dfrac{\prod_{j=0}^{n_2-1}\rho_2(j)}{n_2!b_{min}^{n_2-H}b_{max}^{L}} & for\ c-m < n_2 \le C/b_{min} \end{cases} \tag{7.3}$$

It can be verified that the stationary system satisfies Kolmogorov's Criteria for reversibility.

Without causing ambiguity, the term "call" in this chapter is used for both burst and call. When the population is finite, the "call" should be understood as burst.

7.3 Optimal Policies Structure

7.3.1 Coordinate Convex Policies

Most studies on optimal resource access consider a special class of access control policies, namely, the *coordinate convex* policies [18], [19], [27], [36], [60].

Definition 7.1 *A policy associated with state set Ω is said to be coordinate convex (CC) if $\mathbf{n} \in \Omega$ and $n_i > 0$ implies $\mathbf{n}_i^- \in \Omega$ for all class-i, and a class-i call is accepted when the system is in state $\mathbf{n}$ if and only if $\mathbf{n}_i^+ \in \Omega$.*

The CC policies are a very rich class of policies, since the only requirement is to allow state transitions in both directions along each coordinate. The state transition from $\mathbf{n}$ to $\mathbf{n}_i^-$ naturally must be allowed, otherwise it would mean that no class-i calls can be completed when the system is in state $\mathbf{n}$. The actual requirement, thus, is to allow the state transition from $\mathbf{n}_i^-$ to $\mathbf{n}$ if $\mathbf{n}$ and $\mathbf{n}_i^+$ are both in the state set Ω. This a practical condition for broadband communication networks, as a relatively small decision table is required for all possible states, as compared to for all possible state transitions. This is also a reasonable condition. Because the calls from the same class are considered to be statistically identical, there is no particular reason that the n_1 class-1 calls and n_2 class-2

calls at state $\mathbf{n}$ would behave differently if state $\mathbf{n}$ is reached from different paths. Therefore, it is reasonable and practical not to limit direction of state transition between two neighboring states.

The discussion on access control policies in this thesis is limited to coordinate convex policies only. Henceforth, the term "policy" will mean coordinate convex policy.

A CC policy can be identified by a set of states, and, hence, notations of set theory are used here to refer to the associated stochastic processes in the following discussion.

The associated stochastic system for any policy is a truncated process of the reversible process for Ω_F. Following the property of truncated reversible processes (Corollary 1.10 in [38]), the equilibrium probabilities can be obtained.

Lemma 7.1 *The equilibrium probability for the stochastic process under policy Ω being in state $\mathbf{n}$ is denoted by $p_\Omega(\mathbf{n})$. Then,*

$$p_\Omega(\mathbf{n}) = q_1(n_1)q_2(n_2)/G(\Omega) \tag{7.4}$$

where $q_1(n_1)$ and $q_2(n_2)$ are those defined for Ω_F by equations (7.2) and (7.3), and the normalization factor $G(\Omega) = \sum_{\mathbf{n}\in\Omega} q_1(n_1)q_2(n_2)$

7.3.2 Reward Function

The objective of optimal access control is to maximize the reward. More specifically, if the time axis is finite, the objective can be translated into optimization of the total reward during that period of interest. The time period is, however, often not known, or the time axis is infinite without a future cutoff point. In this case, the objective of optimal access control then becomes maximizing the average reward.

Let column vector $\mathbf{r} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$ be the reward rate per call for each of the classes. The unit of reward is chosen such that $r_i \geq 1$ for all i. The average reward for a policy Ω is denoted by $R(\Omega)$, and

$$R(\Omega) = \sum_{\mathbf{n}\in\Omega} \mathbf{n}\mathbf{r}p_\Omega(\mathbf{n})$$

The optimal access control policy then can be defined for a given reward function.

Definition 7.2 *A policy Ω_{op} is said to be optimal if*

$$R(\Omega_{op}) = \max\{R(\Omega) : \Omega \subseteq \Omega_F\}$$

Lemma 7.2 *Let a, b, c and d be constant integers such that $a \leq b$ and $c \leq d$. If $\Omega = \{(n_1, n_2) \in \Omega_F : a \leq n_1 \leq b, c \leq n_2 \leq d\}$, Then*

$$R(\Omega) = r_1 x_1(a, b) + r_2 x_2(c, d)$$

where $x_i(a,b) = \frac{\sum_{k=a}^{b} i q_i(k)}{\sum_{k=a}^{b} q_i(k)}$.

Proof.

$$\begin{aligned}
R(\Omega) &= \sum_{i=a}^{b}\sum_{j=c}^{d}(r_1 i + r_2 j)p_\Omega(i,j) \\
&= \frac{\sum_{i=a}^{b}\sum_{j=c}^{d}(r_1 i + r_2 j)q_1(i)q_2(j)}{\sum_{i=a}^{b}\sum_{j=c}^{d} q_1(i)q_2(j)} \\
&= \frac{\sum_{i=a}^{b}\sum_{j=c}^{d} r_1 i q_1(i)}{\sum_{i=a}^{b}\sum_{j=c}^{d} q_1(i)q_2(j)} + \frac{\sum_{i=a}^{b}\sum_{j=c}^{d} r_2 j q_2(j)}{\sum_{i=a}^{b}\sum_{j=c}^{d} q_1(i)q_2(j)} \\
&= \frac{r_1 \sum_{i=a}^{b} i q_1(i)}{\sum_{i=a}^{b} q_1(i)} + \frac{r_2 \sum_{j=c}^{d} j q_2(j)}{\sum_{j=c}^{d} q_2(j)} \\
&= r_1 x_1(a,b) + r_2 x_2(c,d)
\end{aligned}$$

□

The function $x_i(a,b)$ can be considered as the first moment of a birth-death process $X_i(t)$ of states between a and b inclusively. The stochastic process $X_i(t)$ has Poisson arrival of rate $\lambda_i(n_i)$ and has the same departure rate $\mu_i(n_i)$ at state $\mathbf{n}$ as in the Ω_F.

The policy considered in Lemma 7.2 is actually of a rectangular geometry. Two specific cases of it are the following two policies of one dimensional linear geometry. If policy $\underline{\Omega}$ has the form $\underline{\Omega} = \{(i, n_2) : a \le i \le b\}$, then

$$R(\underline{\Omega}) = r_1 x_1(a,b) + r_2 n_2 \tag{7.5}$$

If a policy $\underline{\Omega}$ has the form $\underline{\Omega} = \{(n_1, j) : a \le j \le b\}$, then

$$R(\underline{\Omega}) = r_1 n_1 + r_2 x_2(a,b) \tag{7.6}$$

The following properties of $x_i(\cdot)$ are also used later.

$$x_i(a-1,b) \le x_i(a,b) \le x_i(a,b+1)$$

$$a \le x_i(a,b) \le b$$

The following property of $x_1(\cdot)$ is a direct application of a lemma by Ross and Tsang in [60] for fixed allocation policies.

Lemma 7.3 *[60] For any nonnegative integers a, b, c, d, e, f with $a \le b, c \le d, e \le f, a+c \le e, b+d \le f$; then*

$$x_1(a,b) + x_1(c,d) \ge x_1(e,f)$$

Proof. Since $\mu_1(n_1) = n_1\mu_1$ for the VC level and $\mu_1(n_1) = (N_1 - n_1)\mu_1$ for the burst level, $X_1(t)$ is, thus, the same process considered in Lemma 3 of [60]. Hence, that lemma for fixed allocation policies holds true for class-1, the fixed-call class. □

A particular application of this lemma that will be used later is

$$x_1(0, h(i-1) - h(i)) + x_1(a, h(i)) \geq x_1(a, h(i-1)) \tag{7.7}$$

where $h(i)$ defined in the previous chapter is the maximal number of fixed calls that can be accepted when there are i flexible calls in progress.

This lemma, however, is not applicable to $x_2(\cdot)$, because the condition of the non-increasing $\lambda_2(k)/\mu_2(k)$ cannot be met with flexible calls under FACT.

Lemma 7.4 *If set* $\Omega = \{(a, j-1), (a+1, j-1), \cdots, (b, j-1), (a, j), (a+1, j), \cdots, (b, j)\}$, *and its subset* $\omega = \{(a, j), (a, j-1), (a+1, j-1), \cdots, (b, j-1)\}$, *where* $b > a$ *and* $\Omega \subset \Omega_F$ *(see Figure 7.38). Then* $R(\Omega) > R(\omega)$.

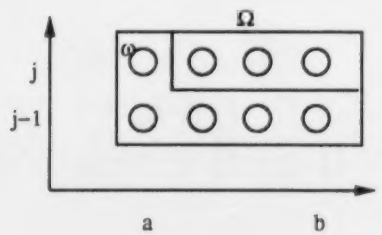

Figure 7.38: Subset Reward Less Than Full Set Reward

Proof. The stochastic process associated with each set is a truncated reversible process of that associated with Ω_F. Thus,

$$\begin{aligned} R(\Omega) &= \frac{\sum_{\mathbf{n}\in\Omega} \mathbf{n}\mathbf{r} q_1(n_1) q_2(n_2)}{\sum_{\mathbf{n}\in\Omega} q_1(n_1) q_2(n_2)} \\ R(\omega) &= \frac{\sum_{\mathbf{n}\in\omega} \mathbf{n}\mathbf{r} q_1(n_1) q_2(n_2)}{\sum_{\mathbf{n}\in\omega} q_1(n_1) q_2(n_2)} \\ &= \frac{\sum_{n_1=a}^{b} \mathbf{n}\mathbf{r} q_1(n_1) q_2(j-1) + (ar_1 + jr_2) q_1(a) q_2(j)}{\sum_{n_1=a}^{b} q_1(n_1) q_2(j-1) + q_1(a) q_2(j)} \end{aligned}$$

Thus, the nominator of $R(\Omega) - R(\omega)$ is

$$
\begin{aligned}
& \sum_{\mathbf{n}\in\Omega} \mathbf{n}\mathbf{r} q_1(n_1)q_2(n_2)\,[\sum_{n_1=a}^{b} q_1(n_1)q_2(j-1) + q_1(a)q_2(j)] \\
& - \sum_{\mathbf{n}\in\Omega} q_1(n_1)q_2(n_2)\,[\sum_{n_1=a}^{b} \mathbf{n}\mathbf{r} q_1(n_1)q_2(j-1) + (ar_1+jr_2)q_1(a)q_2(j)] \\
= & \sum_{\mathbf{n}\in\Omega} \mathbf{n}\mathbf{r} q_1(n_1)q_2(n_2) \sum_{n_1=a}^{b} q_1(n_1)q_2(j-1) - \sum_{\mathbf{n}\in\Omega} q_1(n_1)q_2(n_2) \sum_{n_1=a}^{b} \mathbf{n}\mathbf{r} q_1(n_1)q_2(j-1) \\
& + \sum_{\mathbf{n}\in\Omega} \mathbf{n}\mathbf{r} q_1(n_1)q_2(n_2)q_1(a)q_2(j) - \sum_{\mathbf{n}\in\Omega} q_1(n_1)q_2(n_2)(ar_1+jr_2)q_1(a)q_2(j) \\
= & \sum_{n_1=a}^{b} q_1(n_1)q_2(j-1) \sum_{n_1=a}^{b} [(n_1r_1+(j-1)r_2)q_1(n_1)q_2(j-1) + (n_1r_1+jr_2)q_1(n_1)q_2(j)] \\
& - \sum_{n_1=a}^{b} (n_1r_1+(j-1)r_2)q_1(n_1)q_2(j-1) \sum_{n_1=a}^{b} q_1(n_1)[q_2(j-1)+q(j)] \\
& + q_1(a)q_2(j) \sum_{n_1=a}^{b} [(n_1r_1+(j-1)r_2)q_1(n_1)q_2(j-1) + (n_1r_1+jr_2)q_1(n_1)q_2(j)] \\
& - (ar_1+jr_2)q_1(a)q_2(j) \sum_{n_1=a}^{b} q_1(n_1)(q_2(j-1)+q(j)) \\
= & \sum_{n_1=a}^{b} q_1(n_1)q_2(j-1) \sum_{n_1=a}^{b} [(n_1r_1+r_2)q_1(n_1)q_2(j)] \\
& + q_1(a)q_2(j) \sum_{n_1=a}^{b} [(n_1r_1+r_2)q_1(n_1)q_2(j-1)] \\
> & \ 0
\end{aligned}
$$

Thus, $R(\Omega) > R(\omega)$. □

Definition 7.3 *(a). A nonempty set $S \subset \Omega_F$ is incrementally admissible (IA) with respect to a policy Ω if $\Omega \cap S = \emptyset$ and $\Omega \cup S$ is also a policy.*
(b). A nonempty set $S \subset \Omega_F$ is incrementally removable (IR) with respect to Ω if $\Omega - S$ is also a policy.

Lemma 7.5 *[19] Suppose Ω be optimal, S^- is IR with respect to Ω and S^+ is IA with respect to Ω. Then*

$$R(S^+) \le R(\Omega) \le R(S^-)$$

7.3.3 Threshold Type Policies

Threshold policies have commonly been used in resource allocation to restrict the access of one class, while allowing free access of the others. The formal definition is given below.

Definition 7.4 *(a). A policy Ω is said to be of threshold type-i, i=1,2, if there exists $l_i = 0, 1, \cdots, N_i - 1$ such that*

$$\Omega = \{\mathbf{n} \in \Omega_F : n_i \leq l_i\}$$

where N_i is the maximal number of class-i calls allowed under Ω_F, i.e., $N_1 = C$ and $N_2 = H(0)$.
(b) A policy Ω is said to be of double threshold type if there exist $l_1 = 0, 1, \cdots, C; l_2 = 0, 1, \cdots, H(0)-1$ such that

$$\Omega = \{\mathbf{n} \in \Omega_F : n_i \leq l_i; i = 1, 2\}$$

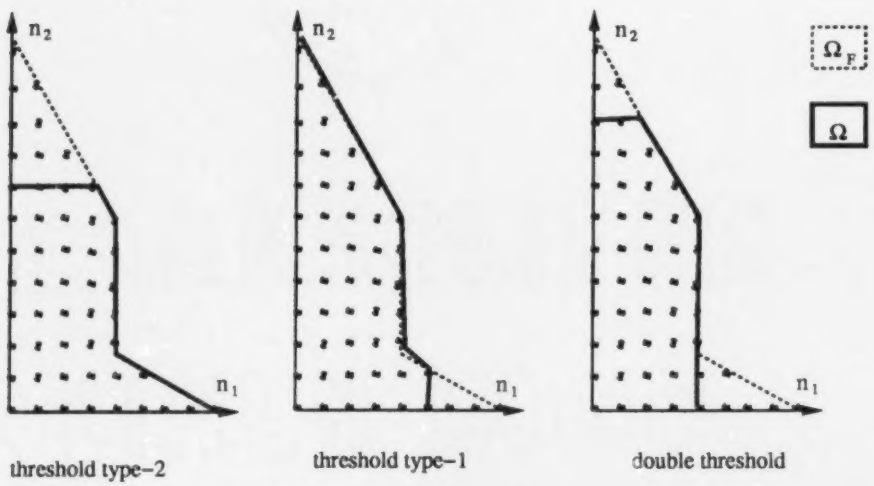

Figure 7.39: Threshold Policies

Figure 7.39 illustrates examples of these three types, where the outer area with dotted boundary lines represents the set of all possible states Ω_F, and the area with solid boundary lines is a policy $\Omega \subset \Omega_F$ of a certain threshold type. Threshold type-1 and type-2 policies restrict, respectively, the access of class-1 (fixed) and class-2 (flexible) calls to the network. Double threshold policies place limits on both classes. It is proved in this section that the optimal policy is in fact of a certain threshold type under some tariff structures.

Definition 7.5 *Let Ω be a policy and $\mathbf{n} \in \Omega_F - \Omega$.*
(a). $\mathbf{n}$ is a double corner point for Ω if $n_1 \geq 1$, $n_2 \geq 1$, $\mathbf{n}_i^- \in \Omega$ for $i = 1, 2$.

(b). $\mathbf{n}$ *is a type-i corner point for* Ω *if* $n_{i^-} \geq 1$, $\mathbf{n}_{i^-}^- \in \Omega$ *and* $n_i = 0$ *or* $\mathbf{n}_i^- \in \Omega$, *where notation* i^- *is the complement of* i, *i.e.,* $i^- = 2$ *if* $i = 1$ *and* $i^- = 1$ *if* $i = 2$.

A double corner point is both a type-1 and a type-2 corner point. In other words, both type-1 and type-2 corner points contain all corner points. As shown in Figure 7.40, a point on the x-axis (point X in the figure) can be a type-1 corner point, but not a type-2; whereas a point on the y-axis (point Y in the figure) can be a type-2 corner point, but not a type-1; and a double corner point cannot be on either axis, according to the definition.

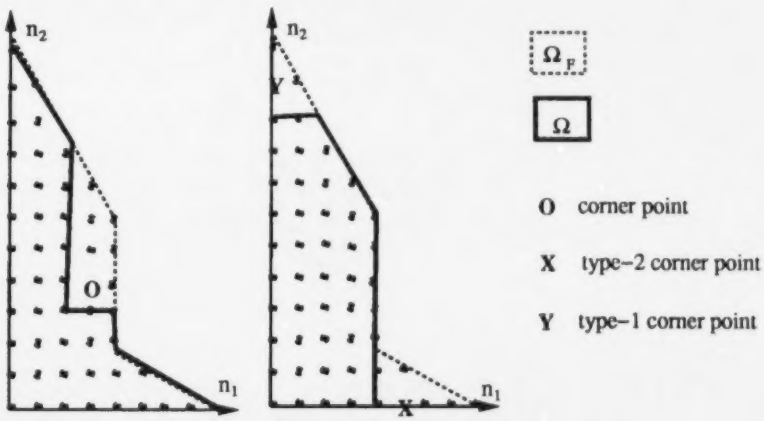

Figure 7.40: Corner Points

From the definition of these three types of corner points, a policy Ω can have a double corner point only if Ω is a true subset of the largest possible set Ω_F; that is $\Omega \subset \Omega_F$. Previous studies allowed $l_i \leq N_i$ in the definition of the threshold types (e.g., [60]), thereby resulting in the largest possible set (Ω_F) itself being all of the three threshold policy types as well. In other existing studies, the existence of a type-i corner point was often assumed during the process of proving a threshold type, implying that the policy under study is a true subset of Ω_F. That is not a valid assumption, if $l_i = N_i$ is allowed. Hence, this thesis has introduced the condition $l_i < N_i$ in Definition 7.4 as opposed to $l_i \leq N_i$.

Lemma 7.6 *[60] (a). A policy is double-threshold if and only if it has no double corner point.*
(b). A policy is type-i threshold if and only if it has no type-i corner point.

The proof of a threshold policy, hence, amounts to proving the nonexistence of the corner points of the corresponding type.

7.3.4 Optimal Policy Structures

Lemma 7.7 *If (n_1, n_2) is a type-2 corner point for policy Ω, then there exists $j = 1, ...H(0)$ such that $n_1 = h(j) + 1$. Moreover, $\frac{r_2}{r_1} \geq \frac{1}{j}$. The functions $H(i)$ and $h(j)$ have been defined in Chapter 6 as the maximal number of class-2 (flexible) and class-1 (fixed) calls that can be served by the network when there are i class-1 and j class-2 calls in progress, respectively.*

Proof. The proof is divided into two steps. The first step uses contradiction to show that there exists a $j = 1, \cdots, H(0)$ such that $n_1 = h(j) + 1$. The second step is to prove that this j further satisfies $r_2/r_1 \geq 1/j$.

First suppose there is no $j = 1, ..., H(0)$ such that $n_1 = h(j)+1$. Let $n = \max\{k : (n_1-1, n_1+k) \in \Omega\}$. By the definition of type-2 corner point, $(n_1 - 1, n_2) \in \Omega$. If we remove $S^- = \{(n_1 - 1, n_2 + i) : i = 0, \cdots, n\}$ from Ω, the remaining set $\Omega - S^-$ is still a policy. In other words, S^- is IR with respect to Ω.

Since $n_1 \neq h(j)+1$ and $(n_1, n_2) \notin \Omega$ by the definition of a type-2 corner point, $S^+ = \{(n_1, n_2+i) : i = 0, ..., n\}$ is IA with respect to Ω. The condition $n_1 \neq h(j) + 1$ comes into play here, because otherwise S^+ may not contain $(n_1, n_2 + n)$. Examples in Figure 7.41 can serve as visual aids.

From equation (7.6) we now have

$$R(S^+) = r_1 n_1 + r_2 x_2(n_2, n_2 + n) \geq r_1(n_1 - 1) + r_2 x_2(n_2, n_2 + n) = R(S^-)$$

which contradicts Lemma 7.5. Therefore, there exists a j such that $n_1 = h(j) + 1$.

Second, now redefining $S^+ = \{(n_1, n_2)\}$ and

$$S^- = \{(x, y) \in \Omega_r : n_2 \leq y \leq j, h(j + 1) + 1 \leq x \leq h(j)\}$$

With respect to Ω, S^+ is IA, and $S^- \cap \Omega$ is IR. The example in Figure 7.41 can serve as a visual aid.

From Lemma 7.2,

$$\begin{aligned} R(S^-) &= r_1 x_1(h(j + 1) + 1, h(j)) + r_2 x_2(n_2, j) \\ R(S^+) &= r_1(h(j) + 1) + r_2 n_2 \end{aligned}$$

And $R(S^-) \geq R(S^- \cap \Omega) > R(S^+)$, according to Lemma 7.4, and Lemma 7.5. Therefore,

$$\begin{aligned} \frac{r_2}{r_1} &\geq \frac{h(j) + 1 - x_1(h(j + 1) + 1, h(j))}{x_2(n_2, j) - n_2} \\ &\geq \frac{h(j) + 1 - x_1(h(j + 1) + 1, h(j))}{j} \\ &\geq \frac{1}{j} \end{aligned}$$

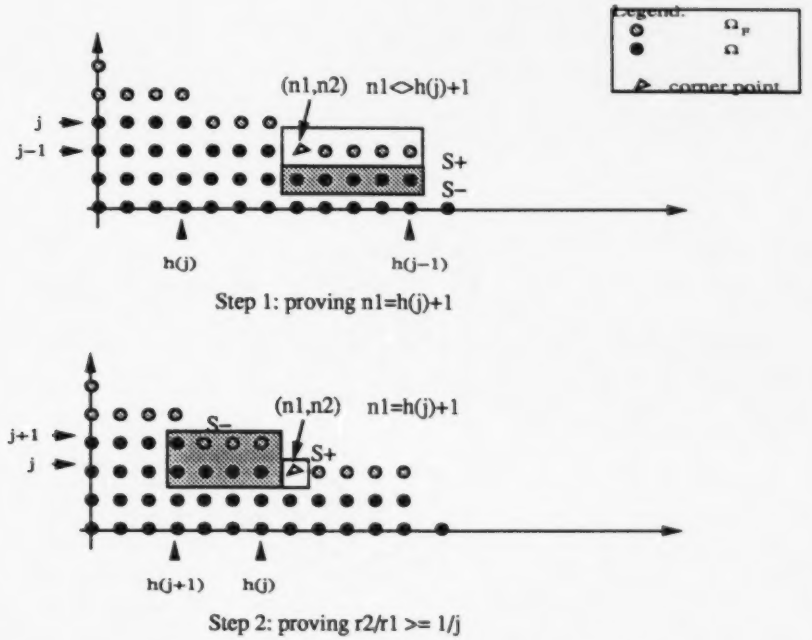

Figure 7.41: Position of Type-2 Corner Point

as $a \leq x_i(a,b) \leq b$. This completes the proof. □

This lemma immediately leads to the following theorem on the relation between tariff structure and the optimal policy structure.

Theorem 7.1 *If the tariff is structured such that profit rate of a flexible call is less than b_{min}/C times that of a fixed call, i.e., $r_2/r_1 < b_{min}/C$, then long term average profit can be maximized by limiting the number of flexible calls allowed to access the link.*

Proof. Suppose that optimal policy Ω_{op} is not of threshold type-2 when $r_2/r_1 < b_{min}/C$. Then, according to Lemma 7.6 Ω_{op} has a type-2 corner point. Furthermore, according to Lemma 7.7, there exists an integer $j = 1, \cdots, H(0)$, such that $r_2/r_1 \geq 1/j \geq 1/H(0) \geq b_{min}/C$. This contradicts the condition $r_2/r_1 < b_{min}/C$. Thus, the optimal policy cannot have a type-2 corner point when $r_2/r_1 < b_{min}/C$. Therefore, according to Lemma 7.6, the optimal policy Ω_{op} is of threshold type-2. □

An immediate result of this theorem is the policy form when $r_2 \leq r_1$. This is a stronger condition, but easier to identify, as they can be found directly from the tariff schedule approved

by the government regulatory agency. The optimal policy can then be derived without having to calculate the cost and the profit rate.

Corollary 7.1 *If the flexible-call tariff is lower than the fixed-call tariff, then long term average profit can be optimized by restricting the number of flexible calls allowed to access the link.*

Proof. Since it costs more to support flexible calls than fixed calls, $r_2 < r_1$. The conclusion immediately follows from Theorem 7.1 □

A similar result for a type-1 threshold policy is proved in the rest of this section. To do so, a lemma is needed for a type-1 corner point like Lemma 7.7 for a type-2 corner point.

Lemma 7.8 *If (n_1, n_2) is a type-1 corner point for optimal policy Ω, then there exists $j = 1, ...H(0)$ such that $n_1 = h(j+1)+1$. Moreover, $\frac{r_2}{r_1} \leq x_1(0, h(i-1) - h(i))$.*

Proof. By the definition of a type-1 corner point, (n_1, n_2) is either a double corner point or $n_1 = 0$. In the latter case, $n_1 = h(H(0)+1)+1$ by the definition of $h(j)$. In the former case, $n_1 = h(k)+1$ with $k = 1, \cdots, H(0)$ according to Lemma 7.7. But k cannot be 1 for a type-1 corner point, because $n_1 = h(1)+1$ implies $n_2 = 0$, contradicting the type-1 corner point definition. Therefore, there exists $j = 2, \cdots, H(0)+1$ such that $n_1 = h(j)+1$ for the type-1 corner point (n_1, n_2).

Denote $n = \max\{k : (n_1 + k, n_2 - 1) \in \Omega\}$ Since $(n_1, n_2) \notin \Omega$ by the definition of type-1 corner point, then $n \geq 0$. Let integer $i > 0$ be such that $h(i) \leq n_1 + n \leq h(i-1)$. Since $n_1 + n \geq n_1 = h(j)+1$, we have $0 < i < j$. If we remove $S^- = \{(n_1 + i, n_2 - 1) : i = 0, \cdots, n\}$ from Ω, $\Omega - S^-$ is still a policy. In other words, S^- is IR with respect to Ω. The set above S^-, but with shorter horizontal length, $S^+ = \{(k, n_2) : k = n_1, \cdots, h(i)\}$ is IA with respect to Ω.

Following equations (7.5) we now have

$$R(S^+) = r_1 x_1(n_1, h(i)) + r_2(n_2)$$

and

$$R(S^-) = r_1 x_1(n_1, n_1 + n) + r_2(n_2 - 1) \leq r_1 x_1(n_1, h(i-1)) + r_2(n_2 - 1)$$

Applying Lemma 7.5, we obtain

$$r_1 x_1(n_1, h(i)) + r_2(n_2) \leq r_1 x_1(n_1, h(i-1)) + r_2(n_2 - 1)$$

Thus,

$$\frac{r_2}{r_1} \leq x_1(n_1, h(i-1)) - x_1(n_1, h(i))$$

According to (7.7), it becomes

$$\frac{r_2}{r_1} \leq x_1(0, h(i-1) - h(i))$$

This completes the proof. □

Theorem 7.2 *Denote $B = \lceil b_{max} \rceil$. If the tariff is structured such that the profit rate for a flexible call is more than $x_1(0, B)$ times that for a fixed call, i.e., $r_2/r_1 > x_1(0, B)$, then long term average profit can be maximized by limiting the number of fixed calls allowed to access the link.*

Proof. Notice that $h(i-1) - h(i) \leq B = \lceil b_{max} \rceil$, and $x_1(0, x)$ is an increasing function of x. Thus

$$\frac{r_2}{r_1} \leq x_1(0, B)$$

This follows immediately from Lemma 7.8. □

Apart from b_{max}, the condition in this theorem is also sensitive to the fixed-call traffic flow parameters λ_1 and μ_1. The following corollary presents a condition insensitive to the arrival/departure process ρ_1.

Corollary 7.2 *If $r_2/r_1 > B$, the long term average profit can be maximized by limiting the number of fixed calls allowed to access the link, regardless of traffic parameters.*

Proof. Because $r_2/r_1 > B \geq x_1(0, B)$ regardless any other parameters. □

A very interesting observation can be made here for the connection between the dynamic allocation and the static allocation optimizations. Because the volume (bandwidth allocation) for a flexible call is always smaller than or equal to $B = \lceil b_{max} \rceil$, the reward to volume ratio for a flexible call is always greater than r_2/B, which in turn is greater than that ratio for a fixed call, r_1, in that corollary, as a fixed call always received fixed bandwidth allocation 1 BU. Corollary 7.2, hence, can be interpreted to mean that it is optimal to allow the flexible call class unlimited access, while limiting access to the fixed call class, when the reward to volume ratio for the flexible calls is larger than fixed calls.

This is very similar to the heuristic knapsack algorithm for static allocation, where the class with the largest reward to volume ratio is allowed unlimited access, and the class with lower ratio are admitted only after the class with higher ratio have already been admitted. This seems to suggest that, from the view point of the average reward over a long period, access control can be optimized by ignoring the traffic parameters. In other words, the static allocation might be able to reach the same optimal decision as the dynamic one over a long period.

7.4 Discussions of Multiple Classes

So far, this thesis has proved that, under certain conditions, the optimal access control policy can still be of threshold type if one of the two classes is flexible. The next step is naturally the expansion into more than two classes.

After various attempts, it became clear that, when there is more than one class of flexible calls, the stochastic process is no longer reversible. To illustrate this property, Figure 7.42 provides a counter-example to the reversibility for multiple flexible classes. In this example, bandwidth allocation for class-1 flexible calls is r BUs at n_1 and 1 at $n_1 + 1$. The process is not reversible according to Kolmogorov's Criteria for reversibility.

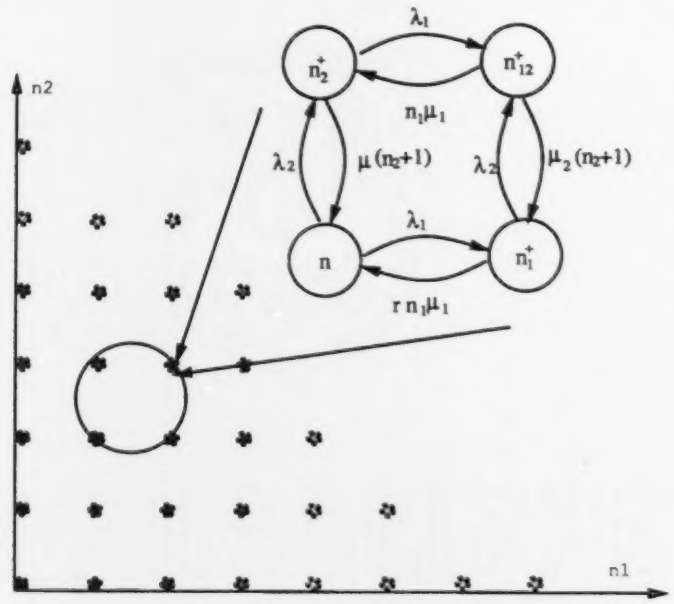

Figure 7.42: Irreversibility Example for Multiple Flexible Classes

This finding indicates that future studies must find a new technique other than the current one with truncated reversible processes in order to prove the existence of a similar threshold structure for multiple flexible classes.

Chapter 8

Simulation Study

A detailed simulator was built to model a communication network with the new flexible bandwidth service, protocols and the FACT scheme. The simulation studies were conducted mainly for the following two reasons.

First, a detailed simulator provides a realistic testbed to evaluate the performance of the flexible bandwidth service, as a complement to the analytical studies. While many assumptions and simplifications are necessary to make the analysis tractable, the simulator can model the network to a very fine detail and under realistic conditions, thereby allowing for the investigation of the impact of various parameters ignored by the analysis. For instance, link transmission latency value is considered to be zero to simplify analysis, but any transmission latency can be easily modeled and studied with a simulator.

Second, simulation also serves as an implementation study for the new service and protocols designed in this thesis. The simulator is actually a detailed implementation of the new service and protocols at both the user equipment and network switches. Many implementation issues can be clarified and verified only with a detailed implementation. Many valuable insights into implementation details have been gained from the process of building this detailed simulator.

The next section provides a brief overview of the simulator. The configuration, traffic and signaling control are presented in Section 8.2. The input control parameters and processing of the captured events are discussed in Sections 8.3 and 8.4, respectively. Section 8.5 addresses the verification and validation of the simulator. The role of simulation in validating the protocol itself is discussed in the last section.

8.1 Overview

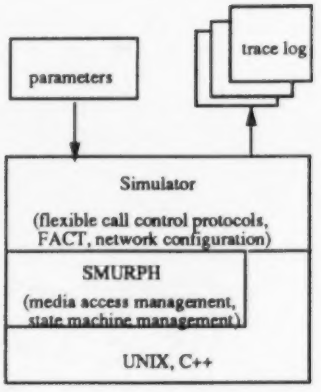

Figure 8.43: Simulation Environment

Figure 8.43 outlines the structure of the simulation environment. The simulator is built with the System for Modeling Unslotted Real-time Phenomena (SMURPH) [21]. SMURPH facilitates controlling of the protocol state machine and media access. In the context of B-ISDN, most functionalities of SONET at the OC level are emulated by SMURPH transparently to the upper layer protocols. The upper layer protocols can access the STS frames directly without having to worry about conversion and transmission at the OC level.

Above SMURPH is the implementation of a communication network with the new service, protocols and the flexible access control technique designed in this thesis.

The simulator produces a log of events which occurred and were captured during the simulation. The basic events recorded are signaling messages and consequent actions taken by the network nodes. The performance statistics are obtained by processing the event log.

The network configuration is controlled from outside the simulator through a set of input parameters. For instance, the comparison of networks with and without FACT is done with the change of one input parameter only.

The granularity of the simulator, or the clock tick in the simulation, was chosen to be the insertion time for one ATM cell at the given STS level. It is the smallest time interval between which two events can be differentiated. For instance, when simulating STS-78, one tick is 108.6ns, or there are about 9.2 million ticks per second. Although granularity at the frame transmission level would be adequate for measuring the performance of signaling protocols, granularity at the cell level was chosen to capture details at an even finer level. To set the simulation tick at the octet level would be unnecessary, however, since most decoding is done at the cell level rather than on an octet-by-octet basis in ATM networks.

Distances in the simulator are measured by the propagation delay in terms of ticks at the media transmission speed. A distance of 1 tick, thus, corresponds to 21.7 meters, if OC-78 is transmitted at 2×10^5 km/s.

8.2 Simulator Configuration

The simulator contains two types of network nodes: switches and local subscriber groups. Traffic originates and terminates at the local subscriber groups, while the switches are responsible for access control, bandwidth allocation and traffic routing. Figure 8.44 illustrates the configuration of the simulation model. At the bottom of the figure is an overview of the simulation model, and at the top is the building module of the simulator with one switch and one local subscriber group.

Since the routing and connectivity problems are outside the scope of this thesis, a network of linear topology is modeled. This topology still permits the study of the situations where traffic enters from and exits to points outside the modeled configuration, because the local subscriber group nodes can be used to model gateways or routers to other networks whose internal details are not of interest to this investigation.

8.2.1 Traffic and Network Nodes

As implied by the name, a local subscriber group node is actually a group of communication service subscribers of the network. The notation "local" is relative to the switch connected directly to that node. Such nodes are also called traffic nodes, because they send and receive traffic streams, or calls.

A call can be of either fixed or flexible bandwidth demand. A traffic node can initiate both flexible calls and fixed bandwidth calls at the same time. It can be considered as containing two camps of subscribers of fixed and flexible bandwidth demands, respectively, with an arriving call from the the first camp with probability mix and from the second with $1 - mix$. The probability for a call being flexible, mix, is specified from the input data file. For the flexible traffic, the acceptable bandwidth range is between $1 - f$ and $1 + f$ BUs, with the flexibility parameter f also defined in the input data file.

```
VPI = (Cell->Receiver == j+d+w )
```

The network nodes are indexed from 0 to $w - 1$ for all switches first, from left to right, then local subscriber groups from w to $2w + 2$, where w is the total number of switches in the network. The left-hand side of a node is said to be side 1, and the right-hand side 2. The traffic node at the side d of switch i thus has index $w + i + d$. Figure 8.45 shows the interconnection and numbering of the network nodes with $w = 4$.

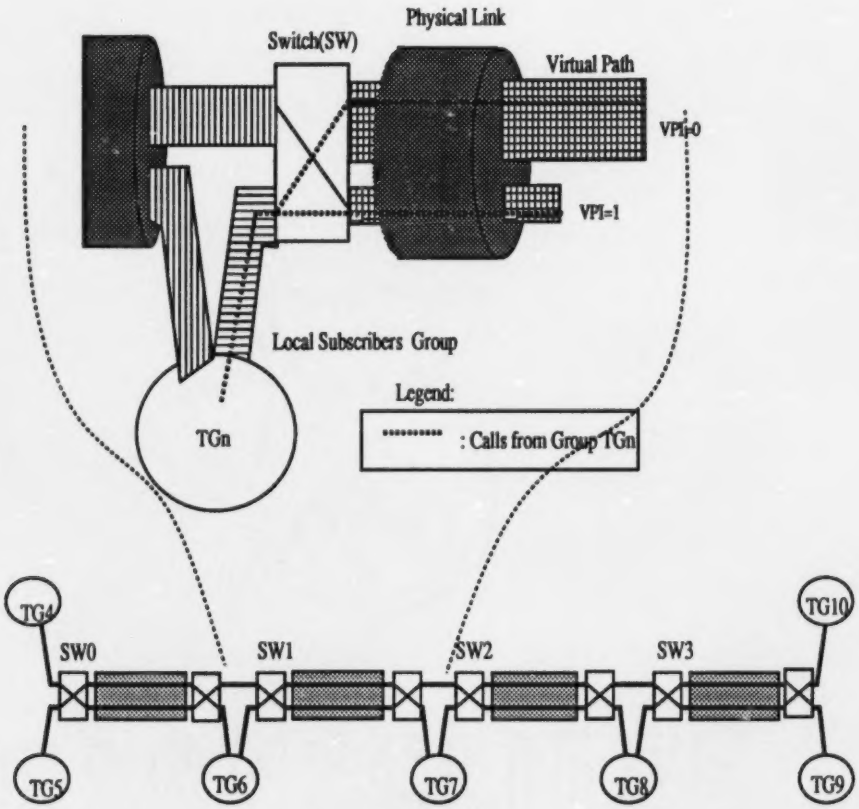

Figure 8.44: Network Model for Simulation

Without loss of generality, a call between two local subscriber group nodes is assumed to be initiated by the node with lower index number. Hence, call requests are always addressed to the nodes located at the right of the calling nodes.

Addressing of the call is randomly distributed such that a call leaving a switch has equal probability of terminating at its local subscriber group or traveling further beyond to some remote nodes. This is accomplished by setting the probability $P_{addr}(i,j)$ for a call from node i being addressed to j, where nodes i,j are not the pairs at the ends of network, as

$$P_{addr}(i,j) = \begin{cases} 0 & i \geq j \\ 2^{-(j-i)} & i < j < 2w+2 \\ 2^{-(j-i-1)} & i < j = 2w+2 \end{cases}$$

This distribution was chosen to achieve an even load at all switches. With load ρ erlangs offered from each traffic node, each input port of any switch, thus, has approximately the same load

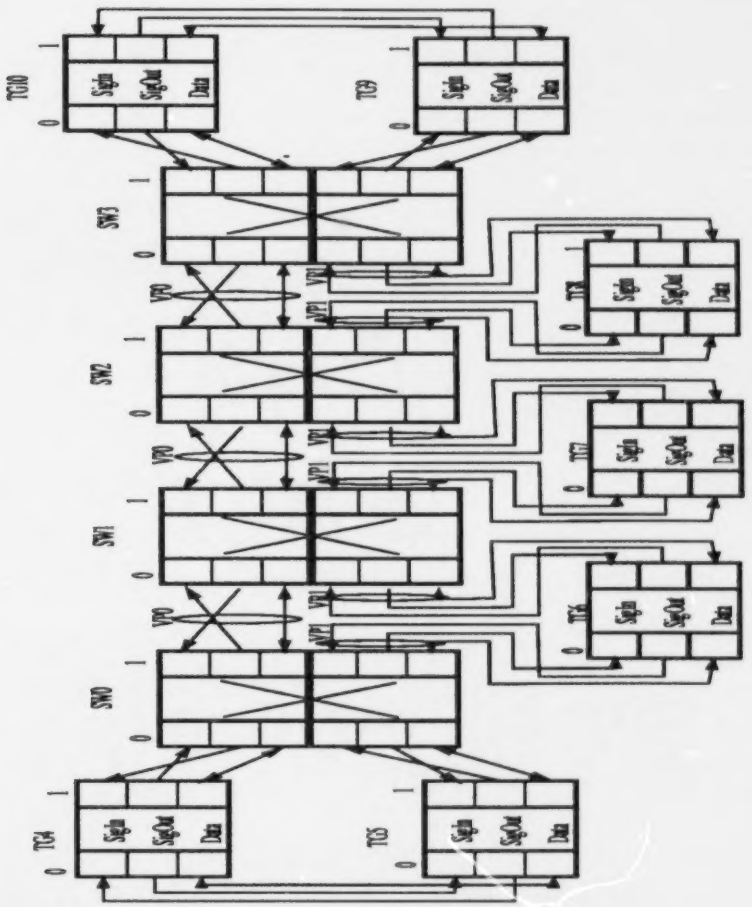

Figure 8.45: Simulator Topology

ρ, when blocking is negligible.

Each switch has two virtual paths at each side, one for transit traffic from/to the remote sites with VPI assigned 0, and the other for local subscriber groups with VPI 1.

The switch routes incoming cells into the output queue of the appropriate VP. Cells are dropped if the queue overflows (and dropped cells are not retransmitted by sources). The routing is accomplished at side d of switch j by a simple VPI assignment to ATM cells in SMURPH as: that is, the VPI is 1 if the cell is addressed to local subscriber group nodes, 0 if the cell is intended to other remote nodes. It is expected that such operation be carried out by hardware in broadband networks. The simulator, hence, assumes this processing delay too small to be distinguished by the simulation

tick.

Switches with newer technology will have to coexist with older switches in future broadband networks. The simulator allows FACT to be deployed at only part of the network switches. This is controlled by an input parameter specifying the percentage of the switches deploying FACT in the simulated network. At the creation of each switch node, tossing of a 100-sided coin is simulated and the outcome compared to that parameter multiplied by 100 to determine whether FACT should be deployed at that particular switch. A parameter value of 1 makes the an across-the-board deployment of FACT, and a value of 0 indicates a network without FACT altogether.

8.2.2 Signaling Protocol

In addition to the conventional call supervision and management tasks, FACT is implemented in the signaling process for bandwidth negotiation and relinquishment. This FACT control can be selected by input data for the convenience of comparing the performance of FACT with the fixed allocation policies.

Calls are generated randomly by the traffic generator in SMURPH according to traffic characteristics specified by the simulation control parameters. Upon a call arrival (generation), a traffic node creates a signal handler for that particular call. The signal handler sends out the call setup request in a SETUP message according to the protocol defined in Chapter 4. This message contains three BC IEs for the maximum, minimum and preferred bandwidth.

The call has to be blocked, if the minimum bandwidth requested is more than the local switch's free capacity plus the relinquishable bandwidth. In this case, a call request denial signal is sent back to the caller, and this call blocking event is captured in the log file.

If the free capacity at the local switch is greater than the maximum bandwidth in the SETUP message, the maximum bandwidth is allocated to that call, and the SETUP message is relayed to the next switch along the route.

Otherwise, the switch attempts to allocate the preferred amount of bandwidth to the call, with relinquishment from in-progress calls first if necessary. A bandwidth value between the maximum and minimum will be assigned to this call. Then, the SETUP message is propagated to the next switch on the route with the preferred bandwidth altered to the bandwidth assigned locally.

The switches will not change the bandwidth allocation of calls originating from remote nodes. They will either accept them or block them. In the event of bandwidth contention, the switches will relinquish bandwidth from the calls originating locally. The relinquishment signaling message is broadcast to the affected parties.

When the SETUP message finally reaches the call destination, the called party also creates

a signaling handler process which responds to the incoming SETUP with a CONNECT message. That CONNECT message will traverse along the same route in the reverse order.

Upon receiving the CONNECT message, that call's signal handler at the calling node creates a data handler process, and records a successful connection in the trace log. The data stream then starts being transmitted by the data handler at the bandwidth indicated in the CONNECT message, and relayed by the switches to the destination.

When the call is completed, the data handler notifies its associated signaling handler, which in turn, releases the bandwidth allocation, terminates both data handler and signaling handler processes of that call, and sends out a DISCONNECT message to tear down the connection. This event is captured in the trace log.

Upon receiving a DISCONNECT message, a switch will examine whether the released bandwidth can be used by its local subscribers, if so the switch re-allocates the bandwidth to them.

8.2.3 Conditions and Assumptions

The following assumptions are made in the simulator.

- The routing is determined by the switches at the beginning of the simulation tick when a cell is received.
- Processing delay for the bandwidth relinquishment and re-allocation is two ticks.
- Cells are processed and transmitted at each node on a FCFS basis. When several cells arrived at the same time, they are processed in a random order (as determined by SMURPH).
- Switches have information of local subscribers only. The network is operated in a distributed fashion.
- The processing time plus insertion time of a signal message is one tick.
- STS-n frames are used without concatenation.
- A call leaving a switch has equal probability of terminating at the local subscribers or of traveling further on.
- Relinquishment is restricted to calls initiated from local subscribers only, and the excessive bandwidth is re-allocated to in-progress calls between local subscribers only.

Although it is possible for a network to extend the effective control region of FACT beyond the local subscribers, the last condition was chosen for the simulation studies for the following reasons. Firstly, if the effective control region were to cover remote subscribers as well, then the re-allocation

request would have to be acknowledged by each node, as the requesting switch does not have up-to-date information of whether the other switches and end users may support the re-allocation. Secondly, the load on the signaling channel would increase drastically if the acknowledgment were to be avoided by bring every switch up to-date on the capability and relinquishment room of all other switches. Thirdly, the call setup time could be as long as a round trip delay across the network when there are other nodes than the direct subscribers involved, as opposed to the one way local broadcast transmission delay. Therefore, it is reasonable and practical for implementation to choose the local switching area as the effective control region for FACT.

8.3 Control Parameters

The first input parameter is the level of STS employed at the link. It is used to derive the simulation tick, the frame size, and payload capacity at each link.

Another parameter defines the number of switches in the network. The simulator itself does not impose any limit on the number of switches; constraints, however, may arise from the memory capacity of the computer on which the simulation is run, and the tolerable length of simulation time. There are two parameters to define the distance between switches, and the distance between a traffic node and its switch. The next parameter is the data queue length at each switch measured by the ratio to the frame length of the given STS level. A ratio 1, for example, indicates that the switch has space to queue one STS-n frame.

The probabilities for FACT being deployed at a switch and for a call being flexible (mix) are also defined in the data file, as well as the flexibility factor (f). These three values must be between 0 and 1, unreasonable input will be rejected by the simulator.

Three pairs of parameters for additional tracing purposes are used. Each pair defines the start and end points of recording a detailed trace of data cells, signal messages, and queueing activities, respectively.

The traffic pattern is defined with three parameters: mean message length in bits of the ATM payload, mean number of arrivals per second, and the preferred call delay. If the preferred call delay is del, and flexible is f, then the acceptable delay for a flexible call is between $del(1-f)$ and $del(1+f)$.

The simulation length is specified with two time limits: the number of seconds of network operation to be simulated, and the simulation execution time. A value zero indicates infinite limit. The simulation will stop when one of the time limits is reached. It took approximately one hour of user time on a SUN-4/40 workstation to simulate two seconds of network operation with one switch.

Usually the time limits should be sufficiently long for the simulated network to be stable. The

Table 8.7: Example of Event Log

Ticks	Location	Signal Type	VCI/Address	Event Tag	Bandwidth
35593545	S02/P018	SETU	61/02-03	CREQ	0.031250, 0.018750, 0.031250
35593556	S00/P003	NOTI	61/03-02	LINQ	-0.000891
35593568	S02/P230	CONN	61/02-03	SUCE	0.029412
36210510	S02/P189	DISC	58/02-03	FINI	0.029412
36210520	S00/P003	NOTI	58/03-02	REAL	0.000891

choice of simulation time is addressed in the next section when discussing the handling of transients in simulation.

8.4 Processing of Captured Events

The output produced by the simulator is an event log. The basic events simulated and captured are the signaling messages exchanged in the network, and consequent actions at the network nodes. The events can be identified by "event tags". Table 8.7 shows an excerpt of the log recording the following events. The first event has a tag CREQ for call request, indicating that Node S02 initiated a call (number 61) to Node S03 at 35593545 ticks during the run, requesting bandwidth between about 0.018 and 0.031 of an STS frame. As indicated by LINQ, the switch (Node S00), noticing bandwidth contention on a virtual path, had to relinquish about 0.0009 of a frame from each in-progress local call on that VP. Tag SUCE represents the successful setup of a call. When the caller finally received the call connection indication, 0.029412 of a frame was allocated to its connection. A while later, the completion of one of the calls between the same calling and called nodes enabled the switch to re-allocate the freed bandwidth to all in-progress calls. The Call 61 was, thus, using $0.029412 + 0.000891 = 0.030011$ of a frame then.

The event log is post-processed to obtain the performance measurements of interest. In other words, the data induction is decoupled from the simulation. The advantage of doing this is to allow for change in measurements and measurement methods without having to go through the time consuming process of modifying, testing and re-running the simulator again.

The post-processing is to provide accurate observations of simulated network activities. The accuracy must be considered in choosing the length of a simulation run as well as methods of post-processing. Three factors can affect the observation accuracy: simulation run length, start and termination transient effects.

Transient effects can occur at the beginning and the end of the simulation. A transient effect at the beginning, namely start transient, occurs before equilibrium has been achieved. At the termination of the simulation, there are still some calls and signaling messages in progress, and no

definitive outcomes of these unfinished activities can be obtained from the simulation. This effect is referred to as terminate transient effect.

The detection of transient periods is not a part of the simulator, but part of the post-processing of the simulation results. With this design, the simulator does not have to make pre-judgment about the purpose of the simulation.

The common approach to identify the transient period at the beginning is to "compute a moving average of the output and to assume steady state when the average no longer changes significantly over time." [17].

It should be pointed out that a network under FACT may still have not reached equilibrium, even when some measures have been stable.

A case in point is bandwidth allocated to each call. Bandwidth allocation fluctuates with the traffic load. Average bandwidth allocation will stabilize when the network is in equilibrium. However, discretion must be exercised when using its stability as an indication of network equilibrium, as it could be misleading; since all calls may get the same maximal bandwidth allocation under FACT for a long period of time after the start of the simulation, especially for a broadband network serving a large number of traffic streams. The term "load-up phase" is used to refer to this period when every call is assigned the maximum bandwidth from the start of the simulation until all the bandwidth is used up. An example is illustrated in Figure 8.46 with the bandwidth allocation for an STS-78 and the average allocation over a moving window of 50 samples. The first 35 million ticks are the load-up phase, where each call is allocated the maximal amount, $b_{max} = 1.25$, bandwidth units.

After links have been loaded up new arrivals can still be accepted by relinquishing bandwidth from the in-progress calls. The start transient is actually comprised of a load-up phase and a "settle-in" phase. The settle-in phase of the start transient is time for the bandwidth allocation of in-progress calls to be reduced from the maximum to an "average" value. The network achieves equilibrium after the load-up and settle-in phases, i.e., the start transient.

During the simulation studies, a very simple, imperative indicator to mark the equilibrium of the system was discovered for FACT. This indicator is the occurrence of the first blocked call, and requires no computation. The rationale for this indicator is the following. No blocking can occur while links are being loaded up. The occurrence of a call blocking event, hence, is a sufficient condition for the network system to have passed out of the transient stage. In other words, the load-up and the settle-in phases have already finished when the first blocking occurs. Thus, the first blocking as an indicator can ensure that the network is already stable. The advantage of this indicator is that no extensive computation and evaluation are required to ensure the passage of the transient period. That observation is verified by numerical analysis of the simulation results. Figure

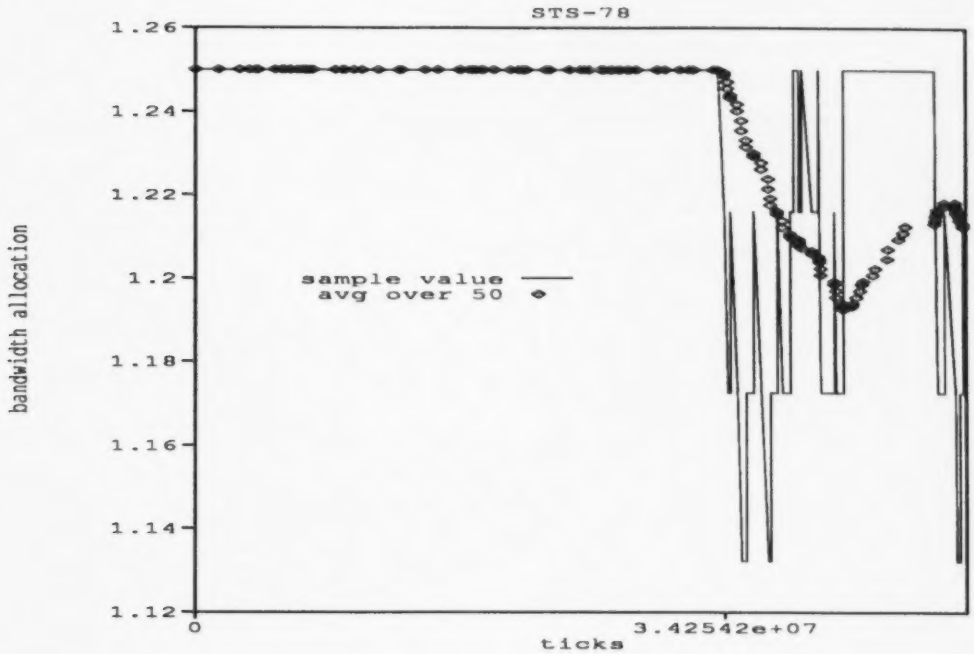

Figure 8.46: Bandwidth Allocation During Load-Up and Settle Phases

8.47 shows the results from two of the simulation runs conducted in these studies with capacities of STS-66 and STS-72. Both confirm that the network equilibrium can be identified successfully by the occurrence of the first blocking event, as supported by the stabilization of call numbers.

The simulation should continue sufficiently long after the network has attained equilibrium in order to ensure accuracy of the statistics. The question now is how long is *sufficient*? The answer can be found by estimating the confidence interval for the population mean bandwidth allocation (e.g. [63], Chapter 5).

The guideline used in the simulation studies conducted for this thesis was obtained as follows. The requirement was to estimate the population mean within an interval of 1/10 of standard deviation (σ) with $(1-\alpha) = 95\%$ confidence. Events were assumed independent first. The confidence interval, thus, has length

$$c = \frac{Z^2_{\alpha/2}\sigma^2}{(\sigma/10)^2} = (19.6)^2 = 384.16$$

where $Z_{\alpha/2}$ is the two-tailed standarized normal statistic for α. Hence, the simulation run should be able to generate at least 385 samples of data after reaching equilibrium. In the simulation studies conducted for the next chapter, for example, it usually took about 20 seconds of operations for the

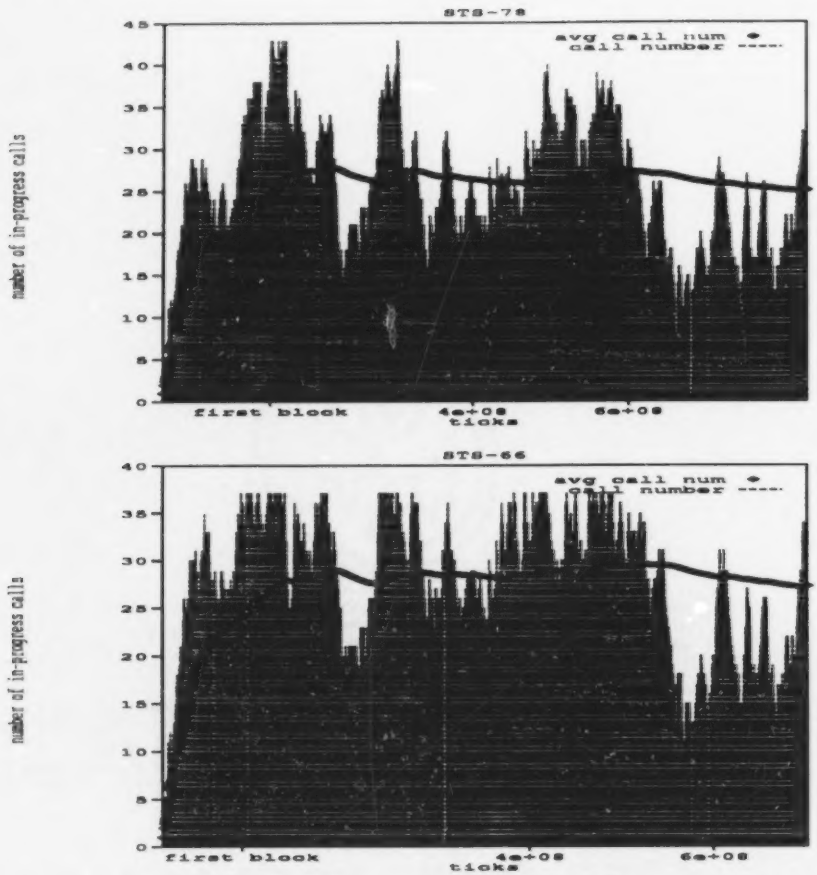

Figure 8.47: First Blocking Indicates the End of Transient Period

simulated network to reach equilibrium as indicated by the first blocking event, then about another 35 seconds to simulate 400 more events. In this case, thus, at least $20 + 35 = 55$ seconds of network operation would have to be simulated.

That 55 seconds limit, however, is based on the assumption that the data samples are independent, while in fact they could be autocorrelated here. One more step was necessary to verify the confidence in the results by considering the autocorrelation after each run. More numerical details are presented in the next chapter.

The transient effect at the end of the run is handled by excluding the activities of unfinished calls or signaling messages from the population. For instance, the sample population for calculating the call blocking rate did not include all the call requests issued during the interval, but all the

requests acknowledged; that is, accepted and blocked calls.

8.5 Simulator Verification and Validation

Confidence has to be established not only for measurements and their analysis as discussed above, but also for the simulator itself. Software is known for its ability to conceal implementation errors. It is important to verify and validate a simulation system so that it can be used with confidence.

Simulator verification usually is to ensure that it behaves as intended, and validation is to determine that "an adequate agreement exists between the entity being modeled and the model for its intended use" [61].

SMURPH and the underlying software system (Unix and C++) are considered reliable and requiring no further verification and validation in this thesis, as both have been used and verified by a large number of users. The suitability of the particular tools provided in SMURPH or its forerunner for telecommunication network simulation has been confirmed in MacGregor's thesis [47]. Hence, for the particular simulations in this thesis, the verification and validation of the upper part of the simulator is of interest, namely, the implementation of the new service and protocols.

The main objective of verification is to show with confidence that the programming is correct. That was done mainly based on the fact that the symptom of incorrect program would be the occurrence of unreasonable or incorrect events according to the protocol design. In addition to debugging and cross-checking results with the analytical model, two steps were taken for dynamic and static verification, respectively. In the first step, various assertion statements were built into the simulator to check unexpected events. Any abnormal events will be detected and reported. Thus, the symptom of incorrectness is checked dynamically each time when the simulator is run. The second step was examining the traces of the protocol state machine execution and signal message flows for conformance with the design.

Validation techniques for simulation models were summarized by Sargent in [61]. They are tabulated here in Table 8.8 with brief descriptions and the check mark ($\surd$) to indicate whether that particular validation technique is applicable for this study. Those applicable technique employed are described below.

Replications are simulation runs with different random number seeds. The internal validity method calls for small result variations among replications. Validation with the traces method and the internal validity method were carried out conveniently with appropriate features of SMURPH [21]. The `-r` and `-t` arguments of the SMURPH command facilitate the selection of different random number seeds and state machine activity tracing, respectively. Additional tracing of the flow and contents of signal messages, data cells and queue status is also implemented in the simulator, and

Table 8.8: Validation Techniques Considered and Applied

Method	Applied	Description
Face Validity	√	review by experts
Traces	√	follow entities through the implementation
Multistage Validation	n.a.	compare outcome to a real system
Internal Validity	√	check variability of several replications
Parameter Sensitivity Analysis	√	compare effect of changing parameters in simulator to that of the analytical model
Comparison to Other Model	n.a.	compare results with other existing models
Historical Data Validation	n.a.	compare results with historical data of real system
Event Validity	n.a.	compare simulated event occurrences with those of a real system
Graphic Display	√	display behaviors graphically as time progresses

controlled through the input data file.

Parameter sensitivity analysis is to change simulator parameters to determine their effect on the model and the output. The same relationships should occur in the analytical model, although the exact values may not be the same (due to the differences in conditions and assumptions). In the simulation studies, four parameters were chosen for sensitivity analysis. The first parameter tested is mix, the percentage of flexible calls among all calls. The simulation results agreed with the theoretical model in that the blocking rate can be reduced by increasing mix. Two traffic parameters in the simulator, average arrival rate and average message length, were expected to have the same effect on the blocking rate as the analytical model predicted for λ and μ, when the average message delay is fixed. It was tested that blocking rate increases as arrival rate or message length increases, when all other parameters are fixed. The last parameter tested for sensitivity is the capacity C. Its increase reduced blocking in both the analytical model and the simulator.

8.6 Protocol Validation Through Simulation

In its broadest definition, protocol validation is to demonstrate that the protocol does what it is intended to do. The validation procedure is to show that the protocol poses certain general desirable properties. The properties commonly checked in protocol validation are the following [74]

- "Protocol should not deadlock,"
- "Protocol should be complete, defining a course of action to be taken in all possible circumstances that can occur when the protocol is executing."

The second property is also referred as the absence of reception errors. Reception errors occur when a message is received in a protocol state where there is no provision for such an incoming message.

In other words, reception errors occur when provisions for some possible events are missing from the protocol.

Automatic protocol validation usually is through dedicated systems (e.g.,[74]). A dedicated validation system is comprised of a communication (network) model, the executable communication protocol, a validation driver and a data base for system states.

Protocol validation through simulation is proposed and applied in this thesis. Simulation usually is considered as a performance study tool, but a properly designed simulator can serve as a protocol validation tool as well. It is like killing two birds – performance study and protocol validation – with one stone, the simulator.

However, a simulator does not automatically qualify as a protocol validation tool. In order to serve as a protocol validation tool, a simulator must provide facilities to demonstrate the two general properties of protocol validation: the absence of deadlock and the absence of reception errors.

The simulator designed in this thesis provides two mechanisms for detecting deadlocks. The first one uses a central registrar for call requests. A call is registered with the central registrar as soon as its signaling protocol starts. The central registrar keeps a record of each call activated for their entire duration. A call is de-registered only after the connection is released by all nodes along the route upon its completion. The central registrar, thus, keeps track of all in-progress calls as well as those calls waiting for establishment or release of connections. The simulator imposes a limit (currently set at 180) on the number of the calls registered with the central registrar at only one time, and will display and record a warning message when that limit is exceeded. A deadlock in the protocol can cause "hogging" at the central registrar by calls in deadlock state, thus the flag for exceeding the limit.

This first mechanism is effective only if deadlocks are widely spread or many calls are involved. The second mechanism provides detection for infrequent deadlocks or deadlocks among a very small number of calls, and is described below.

Deadlock in the simulation may be caused by either a protocol design error or the misrouting of messages in the simulator. Only the first cause is of interest to protocol validation. These two possible causes are distinguished in the simulator by eliminating the possibility of misrouting from a successful simulation run. The simulator vigorously checks for the intended recipient whenever a signal message or data cell is delivered to a traffic node. The simulator will abort, and a misrouting will be declared, if the actual receiver is not the intended one. Thus, protocol deadlock is the only possible cause for an unanswered signal message in a successful simulation run after excluding the terminating period. This mechanism allows for revealing deadlocks by unanswered messages in a successful run.

The absence of reception errors is validated automatically in each simulation run by the assertion statements in SMURPH. Various assertions were built into the protocol implementations at the network nodes to check whether the received message has not been dealt with by the FACT protocol. If the assertions detect that the protocol does not deal with the received message, a protocol error message is displayed and the simulation is aborted. Those assertions can also be viewed as self-diagnosis components of a practical network.

A network designer or operator thus can validate the FACT protocol in two steps. The first step is to run the simulation and take the successful completion of the simulation as an indication of the absence of reception errors and massive deadlocks. The second step is to simply examine the trace log produced by the simulator for any unanswered messages. A short program script can be easily written to accomplish that task.

The practical applications of this protocol validation technique and other results in this thesis are discussed in the next chapter.

Chapter 9

Results and Practical Application

The results of previous chapters are interpreted here through practical scenarios of network design and operation. This chapter demonstrates that the results of this thesis are of practical benefits to communication network designers and operators. These results cannot only be used to improve the network performance, but also provide the network designer and operator with a set of tools to make informed design and operation decisions.

9.1 Network Particulars

The same example of a medical communication network considered in Chapter 3 is used here to illustrate how the work in this thesis can be applied to a practical network design and operation, and how it can be beneficial to the network designer and operator.

The image requests are assumed to arrive according to Poisson processes, and the image sizes to have exponential distributions. The average image size used is 201.3 Megabits, the typical size for many types of radiology images, such as chest radiology, mobile chest, gastro intestinal, and skeletal radiology [11]. A two second delay is a commonly used guideline for interactive transmission. Recognizing that the human users can accept delay variations to some extent, it is further assumed that delay up to 2.5 seconds is acceptable to the users (physicians/radiologists). On the other hand, using excessive bandwidth to accomplish a very short delay is not acceptable to the hospital administration because of its cost. An average delay of no less than 1.5 seconds is assumed to control the communication cost. An average image transmission, thus, requires a service bit rate between 80.52 Mb/s to 134.2 Mb/s. In other words, the preferred service bit rate is 100.65 Mb/s, or one bandwidth unit, and any bit rate between 0.75 and 1.25 BUs is acceptable.

Based on the previous study of medical image communication characteristics [11], the mean arrival rate is assumed to be 14.25 messages per second during peak hours; this is equivalent to

approximately one set of examination results every ten minutes for every radiology staff member in a hospital of more than three hundred beds.

With Poisson arrivals and an exponential image size assumptions, those conditions amount to $\lambda = 14.25$, 1 BU=100.65 Mb/s, $1/\mu = 2$s at BU, and $\rho = \lambda/\mu = 28.5$ for the offered traffic from the application layer.

When ATM is used, however, there are five octets of cell header overhead for every 47 payload octets. The load for interface payload then is $53/47\rho = 32.14$ erlang. The corresponding information payload capacity at the SONET payload level is 113.50 Mb/s, or 2.26 STS-1s.

The distance from switch to the users is taken as 100 ticks. That is equivalent to about 2000 meters for OC-72 at media transmission speed of 2×10^5km/s.

Only broadband image communication is considered in evaluating the network performance, as they constitute most of the bandwidth demands in the virtual private B-ISDN.

The first scenario considers that a new hospital is interested in joining the example virtual private B-ISDN. The second scenario considers the introduction of flexible service into an existing network.

9.2 An Application in Network Design

Suppose a new hospital joins the virtual private B-ISDN. In doing so, its objective is to keep the peak hour call-blocking rate below 3% at the new private B-ISDN node. Furthermore, the hospital wants to maintain the bandwidth utilization as high as possible, and the call delay as low as possible. Calls blocked at the private B-ISDN are rerouted to the public network. In other words, 97% of the calls should be contained within the private B-ISDN, and no more than 3% of calls on average would overflow to the public network.

There are four bandwidth allocation policies that the network designer can choose from. They are:

1. Max – fixed allocation at the maximum bandwidth request 1.25 BUs, this policy provides the shortest delay, but requires the largest amount of bandwidth provisioning;
2. Min – ixed allocation at the minimum bandwidth request 0.75 BUs, this policy requires the least amount of bandwidth provisioning among all fixed allocation policies, but has the longest transmission delay.
3. Avg – fixed allocation at 1 BU, this policy is a tradeoff between the first two fixed allocation policies;

4. Flex – flexible between $b_{min} = 0.75$ and $b_{max} = 1.25$ BUs using FACT.

Policies are chosen by evaluating their impact on the following network design issues:

- Bandwidth Provisioning – how much bandwidth capacity is required at the link/switch to meet the design objective,
- Emergency Handling – if and how the emergency demands (e.g., fiber cut, natural disaster) can be handled, and
- Signaling Capacity Requirement – how much signaling capacity is required to support the policy, especially for the negotiations in FACT. The analysis in Chapter 6 indicates a high frequency of negotiation signaling activities. Thus, a concern that could arise in practical applications is whether or not the signaling channel in the network will be overloaded by signaling traffic of the new service and protocols.

9.2.1 Bandwidth Provisioning

The first task facing the network designer is to determine how much bandwidth is needed to meet the performance requirement with a particular policy. This task can be undertaken in two steps using the results of this thesis.

The first step is to use the analytical model to obtain some preliminary results. At this stage, the impact of signaling transmission and processing delay is ignored, considering that they have the same effect on both fixed allocation and flexible allocation. Many factors are also ignored in the analysis in order to simplify it. The second step is to determine more accurately the relationship between performance and bandwidth provisioning.

Theorems 6.1 and 6.3 tell the network designer that, with the same capacity, FACT can result in a lower blocking and a higher utilization than fixed allocation policies.

More precisely, according to Theorem 6.2, with the same aggregate capacity STS-72 ($C = 30, H = C/b_{min} = 40, L = C/b_{max} = 20$), the blocking rate with FACT is less than $u^{H-C} = 0.95^{40-30} = 59.87\%$ of the blocking rate with a fixed allocation at 1 BU (100.7 Mb/s), and $u^{H-L} = 0.95^{40-20} = 35.85\%$ of that at 0.75 BU (75.53 Mb/s). In other words, the deployment of FACT can result in almost twofold and threefold reductions in blocking rate compared to fixed allocations at the average bandwidth and minimal bandwidth, respectively.

Although that theorem does not provide a bound for the maximum allocation similar to those for the other two fixed allocation policies, the theoretical value of blocking rate can easily be obtained from Equation 6.1 for fixed allocation and from Equation 6.7 for FACT for various bandwidth

provisioning options. Table 9.9 shows the blocking probabilities for $\rho = 32.14, b_{min} = 0.75$, and $b_{max} = 1.25$.

Table 9.9: Blocking Probability (%): FACT versus Fixed Allocation

STS	C	FACT	Min	Avg	Max
68	26.5	4.49	12.46	13.09	15.86
70	27.4	2.89	9.71	11.31	13.55
72	30.8	1.84	8.47	9.66	11.43
74	31.7	1.11	7.31	8.15	9.50
76	32.5	0.57	5.28	6.78	9.50
78	33.4	0.31	4.42	5.56	7.77
80	34.2	0.16	3.65	4.49	6.24
82	35.1	0.07	2.39	3.57	4.92
84	36.0	0.03	1.90	2.79	3.80
86	36.8	0.02	1.48	2.14	3.80
88	37.7	0.01	0.87	1.61	2.88

Since an STS-n without concatenation provides about 96.7% payload with the rest being overhead, $n = 2.26C/0.967$ is used in this table to obtain the STS level n from the capacity value C.

The results indicate that for a 3% blocking rate a payload capacity equivalent to at least STS-82 would be necessary should the fixed allocation be used, while an STS-70 might be adequate for FACT. Hence, the choice of FACT is made over the fixed allocation policies, and the fixed allocation policies are no longer considered.

The bandwidth capacity value obtained from the analytical model is only an approximation. Because of the assumptions such as zero signal processing delay and re-allocation delay and C being an integer multiple of b_{min} and b_{max}, the analytical results tend to underestimate the capacity requirement. The analytical model nevertheless provides a valuable tool to quickly eliminate several inferior policies and to find a capacity range for the selected policy.

The simulation study is then used to verify and fine tune the results under more realistic conditions. The analytical results indicate that simulation should use at least an STS-72.[1] Several simulation runs were carried out for different STS levels (STS-72, STS-78 and STS-84) for 90 seconds of network operations (about four days of simulation time).

First, the number of in-progress calls was analyzed. The occurrence of the first blocking event was used as the indicator of equilibrium for STS-72 and STS-78. But no blocking occurred for STS-84 during 90 seconds of network operation. Thus, the average number of in-progress call was

[1] STS-70 is sufficient according to the analytical result, but STS-72, rather than STS-70, was used because the broadband provisioning usually is carried out in multiples of STS-3s.

used to identify the equilibrium point for STS-84. Figure 9.48 plots the number of call and its mean for STS-84. The results clearly indicate that the system has stabilized after 2×10^8 ticks, or about 20 seconds of network operation.

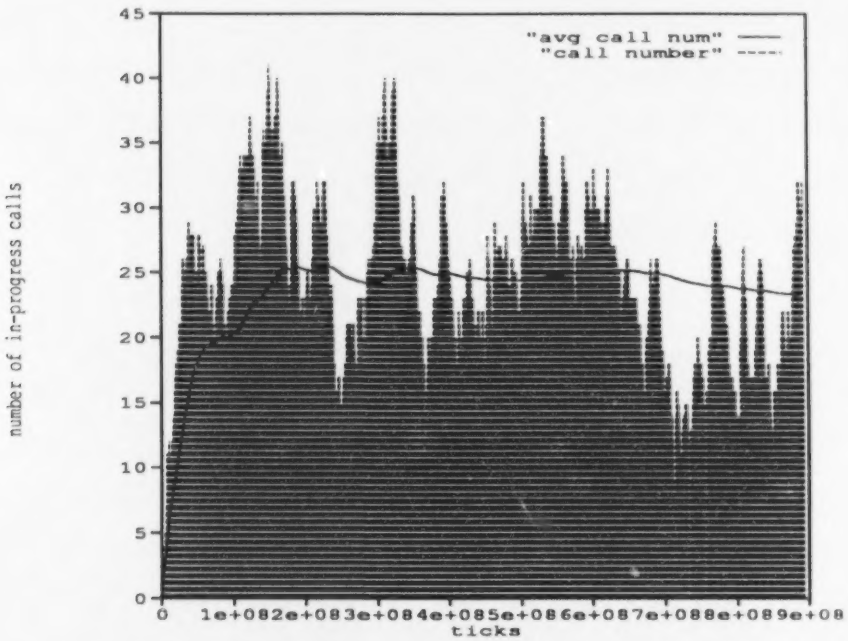

Figure 9.48: Identifying Equilibrium Phase for STS-84 through Number of Calls

The 90 seconds total, or 70 seconds in equilibrium, simulation length was verified for the 95% confidence criterion. For instance for STS-84, a population of 219 samples is required for the mean number of calls to be estimated within 5% of the actual average value with 95% confidence. The pilot sample population used is 50, and the maximal time lag for computing autocorrelations used is 5, or 10% of samples ([63], Formula 5.19). There are on average $\lambda = 14.25$ call arrivals and departures every second. A population of 219 samples corresponds to less than 16 seconds of network operation. Thus, the 90 seconds total simulation time is sufficient for the analysis to be within 5% of the average value with 95% confidence. The same procedure was also applied to the other capacities as well. All results confirmed the sufficiency of the simulation run lengths used.

Table 9.10 shows the mean number of in-progress calls, its standard deviation and maximum number of calls under FACT, as compared to the maximum number under fixed allocation policies.

The phenomenon that a higher number of in-progress calls is observed at a lower capacity can

Table 9.10: Simulation Results: Number of In-Progress Calls

STS	FACT		Maximum Capacity			
	Mean	Standard Deviation	FACT	Min	Avg	Max
78	29.212	39.0241	44	44	33	26
84	25.127	38.059	47	47	35	28

be explained by lower capacity resulting in lower bandwidth allocation to each call, thus in longer holding times for each call. This is clearly confirmed by the bandwidth allocation results plotted in Figure 9.49.

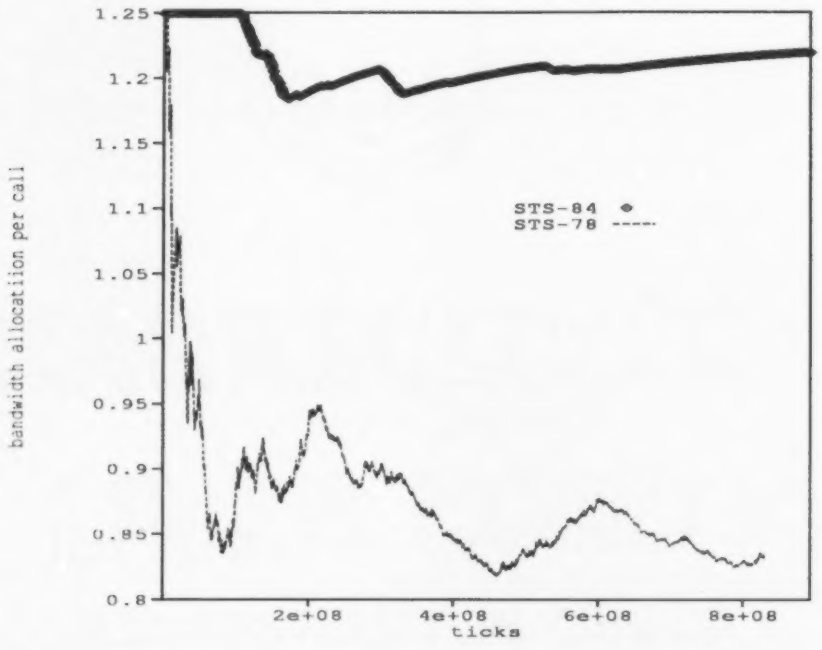

Figure 9.49: Simulation Results: Bandwidth Allocations

The simulation results for average blocking rates are tabulated in Table 9.11 for different bandwidth provisioning options. The call blocking rates were obtained as the ratio of the number of call blocking events to all acknowledged (accepted or blocked) call requests.

Table 9.11: Simulation Results: Call Blocking Rates

ρ	STS	Blocking Rate (%)
28.5	72	2.1
28.5	78	0.6
28.5	84	0.0
30	72	5.4
30	78	1.6

9.2.2 Emergency Handling

One important factor in B-ISDN in general, and in medical communication in particular is the ability to handle sudden surges of network traffic, caused by natural or social disaster such as fiber cuts, earthquakes, riots, tornados and so on.

The emergency capacity, that is the capacity available for emergencies, is defined as the bandwidth that can be used for emergency traffic without disrupting in-progress calls.

With fixed allocations, emergency capacity is limited to the free bandwidth on the link, or the in-progress calls have to be disrupted/preempted. The emergency capacity is simply $C - k$, when there are k in-progress calls with fixed bandwidth allocation at 1 BU each, where C is the total capacity of the link.

FACT provides the emergency handling in a unique way. It allows the bandwidth to be relinquished from in-progress calls with *prior consent* whenever necessary. It can handle the emergency demand gracefully without disrupting the in-progress calls. When there are k flexible calls in progress, $C - kb_{min}$ can be available for emergency traffic. The probability distribution of emergency capacity is given by

$$P(E_c = x) = \begin{cases} p_k & if\, x = C - kb_{min} \\ 0 & otherwise \end{cases}$$

The mean emergency capacity is $C - b_{min}E(k)$, where $E(k)$ is the average number of flexible in-progress flexible calls given by Equation (6.14).

Figure 9.50 depicts the emergency capacity ($C - kb_{min}$) at different simulation instants for STS-78. There are about 12 BUs on the average, and 1 BU at the minimum, available for emergency traffic during the simulation period.

9.2.3 Signaling Capacity

FACT outperforms the fixed allocation policies not without cost. Additional signaling traffic is incurred by bandwidth relinquishment and re-allocation. This cost has to be investigated to determine

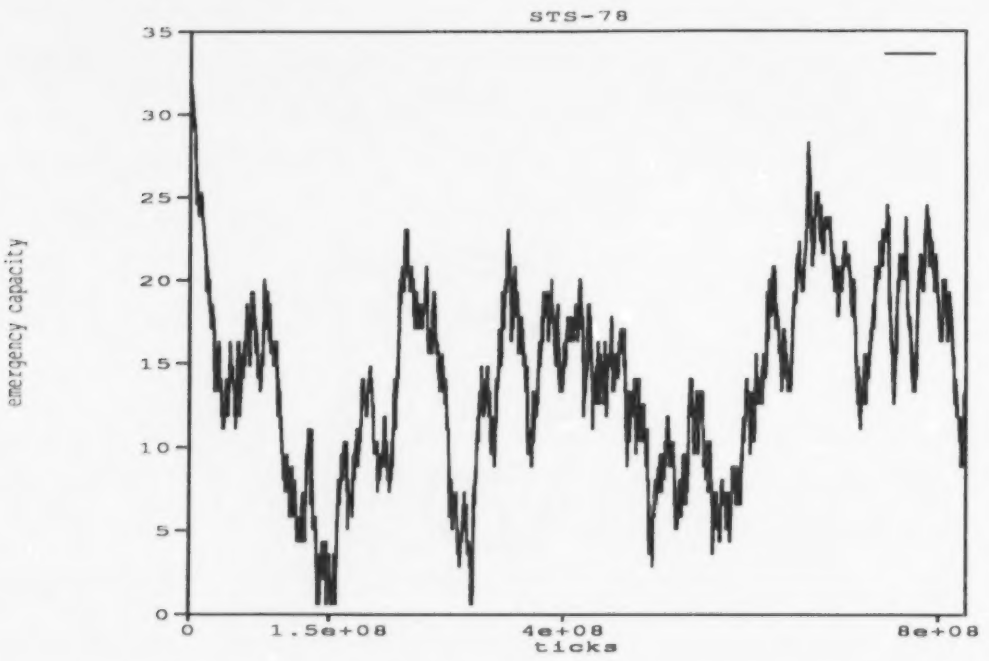

Figure 9.50: Capacity Available to Emergency Demands

whether or not the signaling capacity is adequate.

The event log of the simulation was processed to obtain the load of signaling traffic caused by FACT. The average frequencies of relinquishment and re-allocation, are given in Table 9.12 for various bandwidth provisioning options. The signaling rate is the bandwidth required to carry the

Table 9.12: Signaling Load Investigation

STS	Negotiation Frequency (messages/s)	Signal Rate (Kb/s)
66	18.9	1.51
72	17.8	1.43
78	16.7	1.30
84	13.4	1.07

signal messages for relinquishment and re-allocation. The message length is assumed to be 10 octets. For instance, the signal messages for STS-78 was measured as 16.7 messages/second, or 1.30 Kb/s. As presented in Chapter 2, the signaling capacity suitable for bandwidth relinquishment and re-allocation in all SONET equipment excluding re-generators is $n576$ Kb/s at STS-n (line overhead).

An STS-72, thus, has the signaling capacity of 41.47 Mb/s. Thus, the additional signaling load incurred by the new scheme is well within an acceptable range, and is indeed a negligible price to pay for the significant saving in bandwidth provisioning and improved emergency handling.

9.3 An Application in Network Operation

Suppose after recognizing the benefits the new flexible access control technique has brought to the new site, the network operators at other sites also want to investigate the possible benefits of introducing FACT to their sites and converting the fixed calls into flexible calls. Then two aspects have to be addressed: benefits of upgrading from existing fixed bandwidth allocation service to flexible bandwidth service, and the optimal control of a network with coexistence of both fixed calls and flexible calls.

9.3.1 Benefits of Upgrading to FACT

Based on the estimations in Table 9.9, a hospital with fixed allocation policies must have been using at least STS-82 (with the Minimum Allocation) to maintain a blocking rate below 3%. Now, with FACT the network operator can reduce the bandwidth provisioning by, say, 6% to use an STS-76 instead of STS-82, and still be below a 1% blocking rate.

The benefit is not only in the saving of bandwidth provisioning cost, but also in network performance. While the blocking is maintained at the same level (3%), bandwidth allocated to each call is increased from the minimum 0.75 BUs to close to 1 BU on the average, as indicated by simulation results in Figure 9.49. Consequently, the transmission delay will be reduced by about 25% on average.

The cost of deploying FACT is just a software upgrade at the user equipment and the switches. No new equipment purchase is needed. That makes the new scheme more attractive to network operators.

9.3.2 Optimal Blocking

A virtual private B-ISDN offers a cost effective solution to a community of users. Virtual private networks have only limited capacity, and calls overflow to the public network when the offered traffic exceeds the capacity on a per call basis. It costs more to use the public network than a virtual private network. Thus, private network operators would prefer to optimize the call blocking strategy such that the cost of using the public network is minimized.

The study in Chapter 7 can be applied for this purpose in the following way. Suppose the

calls can be flexible or fixed. Let the rates which the public network charges for a fixed call and a flexible call be r_1 and r_2, respectively. A call accepted at the private network is a call saved from the public network. Then, the problem of minimizing the cost of using the public network amounts to maximizing the saving realized by using the private network.

The two theorems proved in Chapter 7 show that if r_1 and r_2 satisfy certain simple conditions, the optimal control is of threshold type. Therefore, it only takes comparisons of the values C or $1.25C$ to reveal the exact value of the optimal control point. Any call of the controlled class arriving beyond the threshold will be turned away to the public network.

Similarly these two theorems can be used for a public network as well to maximize the long term profit by selectively rejecting call requests.

If the call arrival and departure processes are not Poisson or any known stochastic process, Chapter 5 provides a fast heuristic algorithm to optimally control call admissions and allocate bandwidth on a static basis. The analysis in Chapter 7 suggests that over a long term the static allocation may reach the same control strategy, as the dynamic control with known Poisson arrivals and exponential call length. This was an unexpected yet welcome result.

9.4 Protocol Validation Through Simulation

Last, but not least, is the issue of protocol validation. Before any new protocol is adopted in practical networks, it should pass protocol validation first so that network outage will not be caused by logic flaws in the protocol design.

As the simulation may not cover all possible situations in the network, this validation procedure provides an "indication" only, not a proof of the absence of deadlocks or reception errors. Protocol validation through simulation nevertheless is similar in coverage to the random walk method for a dedicated validation system reported in [74], May, 1992. Confidence in the validation results obtained from any particular automated validation system can be reasonably obtained by running the process for a sufficiently long time. Simulations have to be run sufficiently long for performance studies any way. Thus, no extra cost is incurred by the validation through the simulation approach.

Dedicated protocol validation systems still have their own merits over the validation through simulation approach proposed here. For instance, it is easier to control the testing coverage through a dedicated validation than a simulator. Simulation nevertheless provides network designers and operators with an efficient and inexpensive alternative to a dedicated protocol validation system.

In summary, this chapter applies the results of this thesis in practical scenarios. It shows the benefits, as well as the establishment of confidence, of the new flexible service, protocol and the flexible admission control technique.

Chapter 10

Conclusion

10.1 Summary of Research

A new flexible bandwidth service is proposed in this thesis for broadband communication networks. It is based on the capability of flexible bandwidth provision at the physical level of B-ISDN, general recognition of the need to support bandwidth (re-)negotiation in B-ISDN signaling, and communication users' flexibility in their bandwidth demands.

A flexible-call control protocol is designed and a set of definition documents are provided in this thesis. It includes service primitives, time sequence, PDU definitions and protocol details.

A new access control technique, FACT, is proposed for flexible calls. It is a distributed congestion control scheme. It prevents long term congestion by admission control, and reacts to relieve short term congestion by inducing relinquishment of bandwidth by in-progress calls. It is a distributed control scheme, no global information is required. Control decisions are made at a switch based on local bandwidth allocation and bandwidth status of only those calls that are served by that switch. The control is applied within a certain effective control region of limited radius. Thus, calls that participate in the bandwidth negotiation and reallocation include not only local calls but calls within the same region as well, yet without unacceptable long reaction latency of a network-wide control scheme.

By design, FACT can potentially improve network performance over any fixed allocation policies as well. This possible improvement is studied in this thesis in several ways.

First, if the traffic is an unknown stochastic process, then the static control study shows that FACT always outperforms the conventional fixed allocation scheme. Second, the analysis shows in the form of several theorems that FACT can result in a lower blocking probability and higher utilization than any fixed allocation policies, if the traffic process is Poisson, and no signal processing

delay is considered. Third, simulation studies also confirm the theoretical findings of the advantage of FACT under more general and practical conditions.

The optimal admission control of flexible calls are studied in this thesis for instantaneous (static control) and long term optimization (dynamic control), respectively.

The static admission control and bandwidth allocation problem is modeled as a flexible knapsack problem. The flexible knapsack is a new extension of a general resource allocation model, namely the (static) knapsack problem. The new model allows objectives of the same class to have multiple packing forms, as opposed to conventionally having one form for each class only. Practical instances of the new flexible knapsack include, for example, flexible bandwidth call service, data file compression and merchandise packaging and shipping. Two exact solution algorithms, DP1 and BB1, are designed with dynamic programming and branch and bound methods, respectively. The Algorithm DP1 is less complicated than BB1, but slower. Performance experiments show that Algorithm BB1 is quite fast even for large knapsacks. The empirical results suggest that, for fixed number of classes, its computation time is in the order of C^2, where C is the knapsack capacity.

A heuristic algorithm for solving the flexible knapsack problem is also developed. The performance study shows that it is tremendously faster than the exact solution algorithms. Moreover, the heuristic solutions, on average, are very close to the optimum in most cases, especially when the capacity to average object volume ratio is small. Hence, it can be used as a very efficient approximation for large flexible knapsack problems where this ratio is small.

Two theorems are proved showing that the optimal access control policies are of a certain threshold type when Poisson traffic processes are used. That is an extension of current studies in this area, namely from dealing with fixed calls only into the use of flexible calls. It is proved by several theorems that FACT can always result in lower blocking and higher bandwidth utilization than any fixed allocation policy, under Poisson arrival and exponential call length assumptions.

Furthermore, a holding queue for call requests can be used to drastically reduce the amount of negotiation signaling. The queueing analysis showed that the holding queue is so effective for such purposes that room for holding only one call request can reduce relinquishment signaling by half. However, this queue does not have significant impact on call blocking and utilization. Therefore, such a queue is necessary only if the frequency of relinquishments is a concern.

Although bandwidth negotiations could be frequent at high traffic load according to theoretical analysis, the simulation results showed the signaling message load to be very nominal compared to the signaling capacity provided in B-ISDN.

Simulation is also proposed to be used as an automatic protocol validation tool. Deadlock is detected from the simulation trace by examining any unanswered signal message, when the possibility of misrouting has been eliminated by checking the intended delivery address by recipients. Reception

errors are detected by implementing message type assertions in the protocol for each possible protocol state.

10.2 Contributions and Their Novelty

- identified a new service: the flexible bandwidth call service
- designed a new call control protocol
- proposed a new access control and bandwidth allocation scheme: FACT,
- expanded the knapsack problem to flexible knapsack, and provided both exact and heuristic solutions,
- studied the performance impact of negotiable bandwidth allocation,
- proved the advantage of flexible calls
- proved the threshold structure of optimal access control policies
- constructed a simulator for flexible call service and protocols,
- provided guidelines to practical applications of this new service and protocol, and
- considered the approach of automatic protocol validation through simulation.

10.3 Directions for Future Research

A holding queue of finite length is studied in this thesis. One extension is to further consider limited waiting time. After a certain waiting time, the call request departs from the queue to hunt for the next virtual path or virtual channel.

Even relinquishment is used throughout this thesis. Performance impact of the uneven relinquishment policy still needs to be investigated for the situations where unfairness is not a concern.

The steady state probabilities for more than two classes of flexible calls is an open problem. An even more difficult problem for multiple classes is the analytical proof or disproof of the existence of an optimal access control structure. This thesis has expanded the proof from two fixed classes of previous studies to one fixed plus one flexible class. Similar work for multiple flexible classes would be considerably more difficult than multiple classes of fixed calls, which even now is still an open problem. This thesis shows that the process of two or more classes of flexible calls is generally not reversible. Therefore, if future studies are to find similar threshold structures for multiple flexible classes, they must use a new proof technique that, unlike the current technique, does not rely on the reversibility of the modeled process.

Bibliography

[1] ASATANI, K., HARRISON, K. R., AND BALLART, R. CCITT standardization of network node interface of synchronous digital hierarchy. *IEEE Communications Magazine 28*, 8 (Aug. 1990), 15–20.

[2] BALA, K., CIDON, I., AND SOHRABY, K. Congestion control for high speed packet switched networks. In *IEEE INFOCOM* (1990), pp. 520 – 526.

[3] BALCER, R., EAVES, J., LEGRAS, G., MCLINTOCK, R., AND WRIGHT, T. An overview of emerging CCITT recommendations for the synchronous digital hierarchy: Multiplexers, line systems, management, and network aspects. *IEEE Communications Magazine 28*, 8 (Aug. 1990), 21–25.

[4] BALLART, R., AND CHING, Y.-C. SONET: Now it's the standard optical network. *IEEE Communication Magazine 27*, 3 (Mar. 1989), 8–15.

[5] BERGER, A. W. Performance analysis of a rate-control throttle where tokens and jobs queue. *IEEE Journal on Selected Areas in Communications9*, 2 (Feb. 1991), 165–170.

[6] BULICK, S., AND IRVEN, J. Broadband information services: Concepts and prototype. In *IEEE Globecom'87*, pp. 50.2.1–50.2.8.

[7] CCITT. *Recommendation I.451: ISDN User-Network Interface Layer 3 Specification*, vol. VI - Fascicle VI.11. CCITT, 1988.

[8] CCITT. *Recommendation I.113: Vocabulary of Terms for Broadband Aspects of ISDN.* CCITT, 1991.

[9] CCITT. *Recommendation I.150:B-ISDN ASYNCHRONOUS TRANSFER MODE FUNCTIONAL CHARACTERISTICS.* CCITT, 1991.

[10] CCITT. *Recommendation I.311, B-ISDN General Network Aspects.* CCITT, 1991.

[11] CHANG, T., EID, E. W., MACGREGOR, M., AND MAYDELL, U. Exploratory study of medical image communication systems. *Alberta Telecommunication Research Center Report* (1987).

[12] CHEN, K.-J., AND REGE, K. M. A comparative performances study of various congestion controls for ISDN frame-relay networks. In *INFOCOM* (Ottawa, Ont., Canada, Apr. 1989), IEEE, pp. 674–675.

[13] CIDON, I., AND GOPAL, I. Paris: An approach to integrated high-speed private networks. *Int'l Journal of Digital and Analog Cabled Systems 1*, 2 (April - June 1988), 77–86.

[14] COOPER, C. A., AND PARK, K. I. Toward a broadband congestion control strategy. *IEEE Network Magazine 4*, 5 (May 1990), 18–23.

[15] COOPER, R. B. *Introduction to Queueing Theory*. The MacMillan Company, 1972.

[16] DECINA, M., TONIATTI, T., VACCARI, P., AND VERRI, L. Bandwidth assignment and virtual call blocking in ATM networks. In *IEEE INFOCOM* (1990), pp. 881–888.

[17] EMSHOFF, J. R., AND SISSON, R. L. *Design and Use of Computer Simulation Models*. The MacMillan Company, New York, 1970.

[18] FOSCHINI, G. J., AND GOPINATH, B. Sharing memory optimally. *IEEE Trans. on Communications COM-31*, 3 (Mar. 83), 352–360.

[19] FOSCHINI, G. J., GOPINATH, B., AND HAYES, J. F. Optimum allocation of servers to two types of competing customers. *IEEE Trans. on Communications COM-29*, 7 (July 1981), 1051–1055.

[20] GALLASSI, G., RIGOLIO, G., AND VERRI, L. Resource management and dimensioning in ATM networks. *IEEE Network Magazine 4*, 5 (May 1990), 8–17.

[21] GBURZYNSKI, P., AND RUDNICKI, P. *The SMUPRH Protocol Modeling Environment*. University of Alberta, June 1991.

[22] GERLA, M., AND PAZOS-RANGEL, R. Bandwidth allocation and routing in ISDN's. *IEEE Communication Magazine 22*, 2 (Feb. 1984), 16–26.

[23] GERSHT, A., AND LEE, K. J. Virtual-circuit load control in fast packet-switched broadband networks. In *Globecom* (1988), pp. 214–220.

[24] GERSHT, A., AND LEE, K. J. A congestion control framework for ATM networks. In *IEEE INFOCOM'89* (1989), pp. 701–710.

[25] GILMORE, P., AND GOMORY, R. The theory and computation of the knapsack problem. *Operations Research 14* (1966).

[26] GOLESTANI, S. J. Congestion-free transmission of real-time traffic in packet networks. In *IEEE INFOCOM* (1990), pp. 527 – 536.

[27] GOPAL, I. S., AND STERN, T. E. Optimal call blocking policies in an integrated services environment. In *Conference on Information Sciences and Systems* (1983), The Johns Hopkins University, pp. 383–388.

[28] HALSALL, F. *Data Communications, Computer Networks and OSI.* Addison-Wesley Publishing Co., 1988.

[29] HONG, D., AND SUDA, T. Congestion control and prevention in ATM networks. *IEEE Network Magazine 5*, 4 (July 1991), 10–16.

[30] HU, T. *Integer programming and network flows.* Addison-Wesley, 1969.

[31] HUI, J. Y. Resource allocation for broadband networks. *IEEE Journal on Selected Areas in Communications 6*, 9 (Dec. 1988), 1598–1608.

[32] ILYAS, M., AND MOUFTAH, N. Performance evaluation of congestion avoidance in broadband ISDNs. In *IEEE ICC* (1990), pp. 889–895.

[33] JAIN, R. Divergence of timeout algorithms for packet retransmission. In *Proc. 5th Annual Int'l Phoenix Conf. on Comp. and Commm.* (1986), pp. 174–179.

[34] JAIN, R. Congestion control in computer networks: Issues and trends. *IEEE Network Magazine 4*, 5 (May 1990), 24–30.

[35] JAIN, R. Myths about congestion management in high-speed networks. Tech. Rep. DEC-TR-726, Digitial Equipment Co., Oct. 1990.

[36] KAUFMAN, J. S. Blocking in a shared resource environment. *IEEE Transactions on Communications COM-29*, 10 (Oct. 1981), 1474–1481.

[37] KAWARASAKI, M., AND JABBARI, B. B-ISDN architecture and protocol. *IEEE Journal on Selected Areas in Communications 9*, 9 (Dec. 1991), 1405–1415.

[38] KELLY, F. P. *Reversibility and Stochastic Networks.* John Wiley & Sons Ltd., 1978.

[39] KLEINROCK, L. *Queueing Theorey*, vol. 1. John Wiley & Sons, 1975.

[40] KRAIMECHE, B., AND SCHWARTZ, M. Circuit access control strategies in integrated digital networks. In *IEEE INFOCOM'84* (1984), pp. 230–235.

[41] KRAIMECHE, B., AND SCHWARTZ, M. A channel access structure for wideband ISDN. *IEEE Journal of Selected Areas in Communications 5*, 8 (Aug. 1987), 1327–1335.

[42] LAI, W. S. Frame relaying service: an overview. In *INFOCOM* (Ottawa, Ont., Canada, Apr. 1989), IEEE, pp. 668–763.

[43] LEE, K., AND LIM, Y. Performance analysis of the congestion control scheme in signaling system no. 7. In *INFOCOM* (Ottawa, Ont., Canada, Apr. 1989).

[44] LI, S. R. Algorithms for flow control and call set-up in multihop broadband ISDN. In *IEEE INFOCOM* (1890), pp. 889–895.

[45] LIPPMAN, A., AND BENDER, W. News and movies in the 50 megabit living room. In *IEEE Globecom'87*, pp. 50.1.1–50.1.6.

[46] LOVRICH, A., AND REIMER, J. A multi-rate transcoder. In *IEEE International Conference on Consumer Electronics* (1989), pp. 18–19.

[47] MACGREGOR, M. H. *The Self Traffic-Engineering Network*. PhD dissertation, University of Alberta, Department of Computing Science, 1991.

[48] MALY, K., , OVERSTREET, C. M., QIU, X.-P., AND TANG, D. Dynamic bandwidth allocation in a network. *ACM SIGCOM'88 Symp.* (1900), 13–24.

[49] MARTELLO, S., AND TOTH, P. Branch and bound algorithms for the solution of the general unidimensional knapsack problems. *Advances in Operations Research* (1977), 295–301.

[50] MARTELLO, S., AND TOTH, P. A note on the Martello-Toth algorithm for one-dimensional knapsack problems. *European Journal of Operational Research 20* (1985), 117.

[51] MARTELLO, S., AND TOTH, P. Algorithms for knapsack problems. *Annals of Discrete Mathematics 31* (1987), 312–258.

[52] MINOUX, M. *Mathematical Programming Theory and Algorithms*. John Wiley and Sons Ltd., 1986.

[53] NAGLE, J. On packet switches with infinite storage. *IEEE Transcations on Communication COM-35*, 4 (Apr. 1987), 435–438.

[54] OHTA, S., SATO, K., AND TOKIZAWA, I. A dynamically controllable ATM transport network based on the virtual path concept. In *Globecom* (1988), pp. 1272–1276.

[55] OKADA, T., OHNISHI, H., AND MORITA, N. Traffic control in asynchronous transfer mode. *IEEE Communications Magazine 29*, 9 (Sept. 1991), 58–62.

[56] PATTAVINA, A. Multichannel bandwidth allocation in a broadband packet switch. *IEEE Journal on Selected Areas in Communications 6*, 9 (Dec. 1988), 1489–1499.

[57] PIRKUL, H., AND NARASIMHAN, S. Efficient algorithm for the multiconstraint general knapsack problem. *IIE Transcations 18*, 2 (june 1986), 195–203.

[58] RAMAMURTHY, G., AND DIGHE, R. S. Distributed source control: A network access control for integrated broadband packet networks. In *IEEE INFOCOM* (1990), pp. 896–907.

[59] RASMUSSEN, C., SORENSEN, J., KVOLS, K. S., AND JACOBSEN, S. B. Source-independent call acceptance procedures in ATM networks. *IEEE Journal on Selected Areas in Communications9*, 3 (Apr. 1991), 351–358.

[60] ROSS, K. W., AND TSANG, D. H. The stochastic knapsack problem. *IEEE Transactions on Communications 37*, 7 (July 1989), 740–747.

[61] SARGENT, R. G. Verification and validation of simulation models. In *Progress in Modelling and Simulation*, F. E. Cellier, Ed. Academic Press, New York, 1982, chapter 9, pp. 159–169.

[62] SCHWARTZ, M. *Telecommunication Networks: Protocols, Modeling and Analysis.* Addison-Wesley, 1987.

[63] SHANNON, R. E. *Systems Simulation: the art and science.* Prentice-Hall Inc., Englewood Cliffs, NJ, 1975.

[64] SIDI, M., LIU, W.-Z., CIDON, I., AND COPAL, I. Congestion control through input rate regulation. In *IEEE Globecom* (1989), vol. 3, pp. 1764–1768.

[65] SMITH, D. R., AND WHITT, W. Resource sharing for efficiency in traffic systems. *BSTJ 60*, 1 (Jan. 81).

[66] TANENBAUM, A. S. *Computer Networks*, 2nd ed. Prentice-Hall, 1988.

[67] TIMMS, S. Broadband communications: The commercial impact. In *Proceedings of IEEE INFOCOM'89* (Ottawa, Ont., Canada, Apr. 1989), vol. 2, IEEE, pp. 602–616.

[68] TOWLER, F., ET AL. A 128k 6.5ns access /5ns cycle CMOS ECL static RAM. In *IEEE International Solid State Circuits Conference* (1989), pp. 30–31.

[69] TURNER, J. New directions in communications (or which way to the information age?). *IEEE Communication Magazine 24*, 10 (Oct. 1986).

[70] WALTERS, S. M. BISDN: Flexible bandwidth for the public networks. *Bellcore Digest of Technical Information 8*, 1 (Apr. 1991), 1–11.

[71] WALTERS, S. M. A new direction for broadband ISDN. *IEEE Communications Magazine 29*, 9 (Sept. 1991), 39–42.

[72] WANG, W., SAADAWI, T. N., AND AIHARA, K. Bandwidth allocation for ATM networks. In *IEEE ICC* (1990), pp. 439–442.

[73] WEINRIB, A., AND GOPAL, G. Limited waiting: An adaptive overload-control strategy for circuit switched networks. *IEEE Journal on Selected Areas in Communications9*, 2 (Feb. 1991), 157–164.

[74] WEST, C. H. Protocol validation – principles and applications. *Computer Networks and ISDN Systems 24*, 3 (May 1992), 219–242.